UNIVERSITY OF CALIFORNIA
RIVERSIDE

Physics-Based Electromigration Modeling and Analysis and Optimization

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

by

Zeyu Sun

March 2020

Dissertation Committee:

Professor Sheldon X.-D. Tan, Chairperson
Professor Daniel Wong
Professor Kambiz Vafai

The Dissertation of Zeyu Sun is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

First and foremost, I would like to express the deepest appreciation to my adviser Dr. Sheldon X.-D. Tan for his constant guidance and tireless encouragement that help me successfully complete this work.

I would also like to thank all the committee members Prof. Daniel Wong and Prof. Kambiz Vafai for their beneficial direction, discussion and advises for improving my research.

Also, I would like to extend my special thanks to all my lab members and friends who have helped me and encouraged me during my study, research and life at UC Riverside: Chase Cook, Han Zhou, Hengyang Zhao,Sheriff Sadiqbatcha, Shuyuan Yu, Taeyoung Kim, Yibo Liu.

Last but not least, I would like to thank my family for their support and unwavering love.

ABSTRACT OF THE DISSERTATION

Physics-Based Electromigration Modeling and Analysis and Optimization

by

Zeyu Sun

Doctor of Philosophy, Graduate Program in Electrical Engineering
University of California, Riverside, March 2020
Professor Sheldon X.-D. Tan, Chairperson

Long-term reliability is a major concern in modern VLSI design. Literature has shown that reliability gets worse as technology advances. It is expected that the future VLSI systems would have shorter reliability-induced lifetime comparing with previous generations. Being one of the most serious reliability effects, electromigration (EM) is a physical phenomenon of the migration of metal atoms due to the momentum exchange between atoms and the conducting electrons. It can cause wire resistance change or open circuit and result in functional failure of the circuit. Power-ground networks are the most vulnerable part to EM effect among all the interconnect wires since the current flow on this part is the largest on the chip. With new generation of technology node and aggressive design strategies, more accurate and efficient EM models are required. However, traditional EM approaches are very conservative and cannot meet current aggressive design strategies. Besides circuit level, EM also need to be thoroughly studied in system level due to limited power and temperature budgets among cores on chip.

This research focuses on developing physical level EM model for VLSI circuits and system level EM optimization for multi-core systems in order to overcome the aforementioned problems. Specifically, for physical level, we develop two EM immortality check methods and a power grid EM check method. Firstly, a voltage based EM immortality analysis has been developed. Immortality condition in nucleation phase can be determined fast and accurately for multi-segment interconnect wires. Secondly, a saturation volume based incubation phase immortality check method has been proposed. This method can further reduce the redundancy in VLSI circuit design by immortality check in multiphase. Furthermore, both immortality check methods are integrated into a new power grid EM check methodology (EMspice) as filter for EM analysis. These filters can accelerate the simulation by filtering out immortal trees so that we only need to do simulation on fewer trees that are mortal. Coupled EM simulation considering both hydrostatic stress and electronic current/voltage in the power grid network will be applied to these mortal trees. This tool can work seamlessly with commercial synthesis flow.

Besides physical level reliability models, system level reliability optimization is also discussed in this research. A deep reinforcement learning based EM optimization has been proposed for multi-core system. Both long term reliability effect (hard error) and transient soft error are considered. Energy can be optimized with all the reliability and other constraints fast and accurately compared to existing reliability management techniques. Last but not least, a scheduling based reliability optimization method for multi-core systems has been proposed. NBTI, HCI and EM are considered jointly. Lifetime of the system can be improved significantly compared to traditional methods which mainly focus on utilization.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

1.1.1 Electromigration

Reliability has become a major design challenge and limiting factor for nanometer VLSI designs. As VLSI technology features are pushed to the limit with each successive generation and with the introduction of new materials and increased current densities to satisfy performance demands. As can be seen in the prediction of International Technology Roadmap for Semiconductor (ITRS) [1], current densities keep increasing and will result serious reliability issues. Among these issues, Electromigration (EM) is projected to be a key reliability issue for current and future VLSI technologies. As technology advances into the sub-10nm regime with FinFET devices, it has been predicted that EM failure will become more significant for interconnects due to increased current density and elevated heating. The EM assessment and verification methods are essential in the design of VLSI circuits to

ensure interconnect circuits especially wires in power grid in do not have functional failure and can function will during its lifetime.

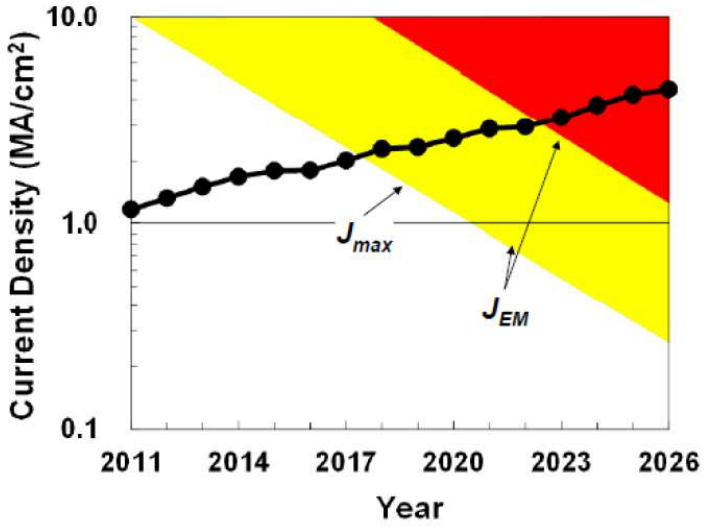


Figure 1.1: Evolution of current densities: J_{max}, the maximum equivalent DC current density and J_{EM}, the current density for targeted lifetime [1]

Accurately predicting the EM-induced time-to-failure for full chip is still a challenging task. Most currently employed EM models are very simple empirical models. They are very conservative and may cause too much redundant in chip chip design. These redundant caused by conservative overdesign and will have extra area, power, and energy costs [3]. Among all these empirical models, Black's equation [4] and Blech limit [5] are the most commonly used. Black's equation use curve fitting method to get the model. In this model, EM induced lifetime only depend on temperature and current densities. This model is very conservative since they do not consider physical meaning of EM and only use experiment data for curve fitting. Also, only single segment with fixed wire width and current density is considered. However, in most of VLSI designs wires are multi-segment

with different current densities. Blech limit is an immortality check method. If the product of current density and length is smaller than a certain critical level, the wire can be considered as immortal. It is generally employed as immortality check method in most designs. However, it also only considers single segment wire and not suitable for modern VLSI design. Besides, all these models only focus on device level EM analysis. System level EM checking method is also required for the high level Mean time to failure (MTTF) check and optimization. In order to overcome the drawbacks of empirical of EM models, many physics based EM models are proposed. Among them one of the most famous model is Korhonen's model [6]. Transient hydrostatic stress described by Korhonen's partial differential equation (PDE) is used to simulate EM effect. However, it is not easy to solve the PDE directly in EDA tools. So, many models and methods are developed to integrate the EM analysis model. A closed form equation derived from Korhonen's equation is proposed in [7, 8]. They get the exact analytic solution of the Korhonen's equation and assume some terms are dominant to calculate EM induced mean time to failure (MTTF). And that method is further improved in [9] with more proper assumption. Finite difference method (FDM) and finite element method (FEM) are also employed to resolve EM problems. For example, [10, 11] use FDM to fast solve the void nucleation and growth process and [12] focus on the post voiding analysis of EM. Comparing with Korhonen's equation, these methods are fast and can be easily integrated to the EDA tools. However, it still takes a long time to do EM analysis with these tools since there are numerous of wires on chip. For these wires, only some of them will fail due to EM effect. So an efficient filter system is required. Also , it is important to combine the EM model with commercial EDA tools.

System level reliability issues also becomes more and more serious with more advanced technology nodes. Due to increasing number of cores and high power density, reliability management methodologies considering reliability issues are essential for lifetime optimization on chips. For system level, reliability issues mainly consists of long term device reliability effects (hard error) and short term transient error of signal (soft error). Hard error such as Negative Bias Temperature Instability (NBTI), Hot Carrier Injection (HCI), and EM, may adversely affect such systems as they decrease allowable executing speed, thereby possibly resulting in unexpected deadline misses. Soft error generally will result to wrong signal or datum and finally result to change instruction or data value. In order to improve the lifetime on system level, many reliability management methodologies are proposed. Work in [13] considers the negative-bias temperature instability (NBTI) and time-dependent dielectric breakdown (TDDB), while [14] mainly focuses on mitigation of the thermal cycling induced reliability at the operation system level. All of those methods did not impacts of soft-errors. Recently, dynamic reliability management for multi-core dark silicons have been proposed [15, 16]. In this work, both EM-based hard errors and soft-errors have been considered. Due to conflicting requirements from both hard and soft errors, the author employs Q-learning as the management method since it can provide cost effective solutions so that the optimal states can be selected to reduce the system energy. However, they are not efficiency enough and lack of online optimization steady. And some scheduling based methods are proposed. [17] uses a simple and general reliability model which cannot represent MTTF in an accurate manner. [18] uses an EM model and focuses on a single-core system. Hence, multi-core systems that are widely used these days cannot

be analyzed and the consideration of only one reliability effect also limits the practicality of this work. [19], which considers multiple reliability effects including EM, time-dependent dielectric breakdown (TDDB), stress migration (SM), and thermal cycling (TC). The work uses temperature to represent such effects, with an observation that they have a strong exponent dependence on temperature. However, temperature is not the only factor that impacts the failure mechanisms. Other factors such as power and operation frequency are also related to these reliability issues; hence, more specific physics-based models are required for the faithful analysis of reliability in modern multi-core systems.

1.2 Dissertation contributions

This dissertation presents advanced technology in the physics-based modeling and system level application of EM. The major contributions of this dissertation are summarized as follows:

- For the voltage based EM (VBEM) analysis we present a novel and fast EM immortality check for general multi-segment interconnect wires. Instead of using current density as the key parameter, the new method estimates the EM-induced steady-state stress in general multi-segment copper interconnect wires based on a novel parameter, *Critical EM Voltage,* $V_{Crit,EM}$. We show that the $V_{Crit,EM}$ is essentially the natural, but important, extension of the *Blech limit* concept, which describes the EM immortality condition for a single segment wire, to more general multi-segment interconnect wires. The proposed method, called voltage-based EM or *VBEM* method, mitigates the problem of current-density-based EM criteria, which can only be applied to a single wire.

The new VBEM method can naturally comprehend the impact of the topology of the wire structure on EM-induced stress. As a result, this new VBEM analysis method is very amenable to addressing EM violations, as it brings new optimization capabilities to the physical design flow. The VBEM stress estimation method is based on the fundamental steady-state stress equations. This approach avoids computationally-intensive numerical methods and can be implemented in CAD tools very easily, as we demonstrate on real design examples. We also show that the proposed VBEM analysis method agrees with results from the finite difference method in the steady state through one example and also agrees with one published closed-form expression of steady-state stress for a special 3-terminal wire case. We also study the impact of current crowding in practical interconnect wires on the estimated steady-state stress, which are shown to be not significant if the length of the wire is much greater than its width. An extension of the VBEM method to consider the significant current crowding effects is also shown and additionally, we analyze mesh-structured interconnect wires and demonstrate that the proposed VBEM method is correct and accurate on such structures.

- For the saturation volume based EM analysis, a new formula of the void's saturation volume for general multi-segment interconnect wires is presented. The new formula is based on the fundamental atom conservation at the steady-state condition of void growth phases. The new void saturation estimation formula agrees with the existing single segment wire saturation void volume formula and is an natural extension of the single segment case to general multi-segment wires. In addition, we consider impacts of

the void volume on final stress distributions of the wire to further improve the accuracy of the proposed formula. The proposed saturation void volume estimation can be applied for fast EM immortality check for nucleated wires, which were considered to be failed wires in the past. Combined with the recently proposed EM immortality check for nucleation [20, 21], we propose a new EM immortality check algorithm, which considers both void nucleation and void growth for the first time and thus overcoming the conservativeness of the existing EM immortality check method. Our numerical results show that the proposed formula agrees well with a published work for two-segment cases, which are supported by experimental data. The formula is also validated by the recently proposed physics-based 3D finite element (FEM) analysis tool for general multi-segment interconnect wires. We also demonstrate new EM immortality check flow can quickly identify the new type of immortal wires, which are nucleated but with smaller-than-critical voids.

- For power grid level EM analysis, we propose a new full-chip EM failure analysis tool, called *EMSpice*, for physics-based coupled EM and electronic analysis. *EMSpice* takes power grid netlists from Synopsys ICC flow, and outputs the failed EM wires and their resistance changes and resulting IR drops of the power grids over the given aging time. It can be used as a EM sign-off tool or used in the EM optimization engine for commercial EDA physical synthesis flow. EM spice can simultaneously considers the hydrostatic stress and electronic current/voltage in a power grid network. The new method starts from first principles and can solve the resulting coupled time-varying partial differential equations in time domain to accurately stress evolu-

tion in multi-segment interconnect trees for EM failure analysis. *EMSpice* simulator employs a finite difference time domain (FDTD) solver for stress analysis for every interconnect tree for both nucleation and post-voiding phases. *EMSpice* simulator also couples the EM analysis with IR drop analysis at time domain. Thus the solver can consider the interplay among stress, void growth, resistance change and IR drop in a single simulation framework. Furthermore, *EMSpice* simulator considers both recently proposed nucleation phase immortality and incubation phase immortality for the first time to remove immortal interconnect trees from EM analysis.

- For machine learning based reliability optimization, we propose a new deep reinforcement learning optimization frame work considering both hard and soft errors for dark silicon processors. Our work is motivated by the lack of online optimization tools and low efficiency of traditional Q-learning algorithm in run time operation. We formulate our DRM problem as minimizing the energy consumption subject to the reliability include hard and soft errors, power, performance and thermal constraints. The hard error model is based on a recently proposed physics based EM model. Hard and soft errors are obtained by recently proposed reliability models. Power, performance are acquired by X86 based micro-architecture model [22] and thermal information is given by temperature simulators. The control operation we use dynamic voltage and frequency scaling (DVFS) and ON/OFF pulsing action. DRL method is used to select the most cost effective operation. Experimental results show that the DRL-based DRM method can lead to energy reduction much better than the Q-learning based method and simple DVFS based method. Furthermore, the DRL-based method also

shows much smaller space complexity (linear versus the quadratic) and less steps to converge than the Q-learning based method as well.

- For scheduling based reliability optimization, we propose a novel resource allocation algorithm to improve the long-term reliability of multi-core real-time embedded systems. To the best of our knowledge, this work is the first to show the dissimilarity between utilization and reliability and propose a reliability-aware task allocation. Utilization is not the only factor related to reliability issue and other factors such as power, temperature and frequency can also hurt the reliability. We combine multiple system-level reliability models, including NBTI, HCI and EM, and apply these models to obtain a trustworthy lifetime estimation of processor cores. Our work is motivated by the fact that conventional allocation methods, such as worst-fit decreasing (WFD), consider only the amount of workload (utilization), which may have a negative impact on lifetime. We show that the proposed algorithm yields significantly benefits in improving system-level lifetime over conventional methods, while guaranteeing timing correctness of all real-time tasks in the system.

1.3 Organization

The rest of this dissertation is organized as follows. Chapter 2 provides the background knowledge about fundamentals of EM. It then focuses on Korhonen's equation based three phase modeling. Chapter 3 presents a voltage based fast EM immortality check for general multi-segment interconnect wires. And a saturation volume based EM immortality analysis is presented in Chapter 4. Chapter 5 presents a new full-chip EM failure analy-

sis tool using filter method and coupled simulation. Chapter 6 focus on machine learning based reliability optimization. Chapter 7 presents scheduling based reliability optimization. Chapter 8 concludes the dissertation.

Chapter 2

Review of three phase based EM physics and stress modeling

2.1 Review of three phase based EM physics and stress modeling

EM is a physical phenomenon of the migration of metal atoms along the direction of the applied electrical field. Atoms (either lattice atoms or defects / impurities) migrate along the trajectory of conducting electrons. Due to momentum exchange between lattice atoms, hydrostatic stress is generated inside the embedded metal wire during migration process. Before the hydrostatic stress reaches the critical level, the atomic flux flowing caused by electron flow from cathode to anode can still balance with the atomic flux caused by inhomogeneous distribution of hydrostatic stress. When the stress reaches the critical level, a void and a hillock formation caused by conducting electrons will be formed at

the cathode end and the anode end. Indeed, when metal wire is passivated into a rigid confinement, which is the case for copper dual damascene structures, the wire volume changes (induced by the atom depletion and accumulation due to migration), and creates tension at the cathode end and compression at the anode end of the line. However, if the hydrostatic stress cannot reach critical level, the void will not be formed due to the balance between atomic flux flowing caused by electron flow and inhomogeneous distribution of hydrostatic stress.

A physics based three phase model is proposed in [9]. EM failure process in this work is described by nucleation phase(t_{nuc}), incubation phase(t_{inc}) and growth phase(t_{growth}). In nucleation phase, stress start accumulate and a void will be formed in the end of nucleation phase. The void start growth in incubation phase. In the end of incubation phase, the cross section of via is blocked by void and can cause either early failure or late failure of the wire. In late failure case, resistance start increase in the growth phase. TTF can be described as the summation of three phases as following:

$$TTF = t_{life} = t_{nuc} + t_{inc} + t_{growth} \tag{2.1}$$

2.1.1 Nucleation phase modeling

Stress evolution is described by the stress-based model Korhonen's model [6]. Transient hydrostatic stress $\sigma(t)$ can be described by Korhonen's partial differential equation (PDE) with the following zero-flux boundary conditions (BC) and initial stress condition (IC) as shown in the following:

$$\frac{\partial \sigma(t)}{\partial t} = \nabla \cdot \left(\frac{D_a B}{k_B T} (\Omega \nabla \sigma - eZ\rho j) \right), \quad \text{in } \Omega_L, \tag{2.2}$$

$$\nabla \sigma(t) = \frac{eZ\rho j_{N_i}}{\Omega}, \quad \text{on } \partial\Omega_L \cap \Gamma_{N_i}, i = 1, ..., k, \tag{2.3}$$

$$\sigma(0) = [\sigma_1(0), \sigma_2(0), ..., \sigma_k(0)] \ , at \ t = 0 \tag{2.4}$$

where $D_a = D_0 \, exp(Ea/k_B T)$ is the effective atomic diffusivity. E_a is the activation energy of the failure process, T is defined as absolute temperature, e is the electron charge, eZ is the effective charge of the migrating atoms, ρ is the wire electrical resistivity, and j is the current density. Ω is the atomic lattice volume. B is the effective bulk elasticity modulus. k_B is the Boltzmann's constant. Ω_L is the domain of simulated copper interconnect and $\partial\Omega_L$ denotes its boundary. Γ_{N_i} is the ith flux termination boundary, where normal current density j_{N_i} are prescribed, among kth termination boundaries. In Eq. (2.2), a uniform D_0, synthesized from different body and boundary diffusion coefficients, is used. Note that Korhonen's hydrostatic diffusion equations can be applied to 3D multi-segment interconnect wires with multiple flux-termination boundary nodes, allowing any number of evolving voids to be simulated. Fig 2.1(a) shows the stress evolution on a wire in the nucleation phase. As shown in the figure, stress at the cathode increase as time goes on. While, stress at the anode will decrease with time.

2.1.2 Incubation phase modeling

When stress reaches a critical level at t_{nuc}, a void is formed. However, the resistance of the interconnect remains almost the same since cross section of the via, which is

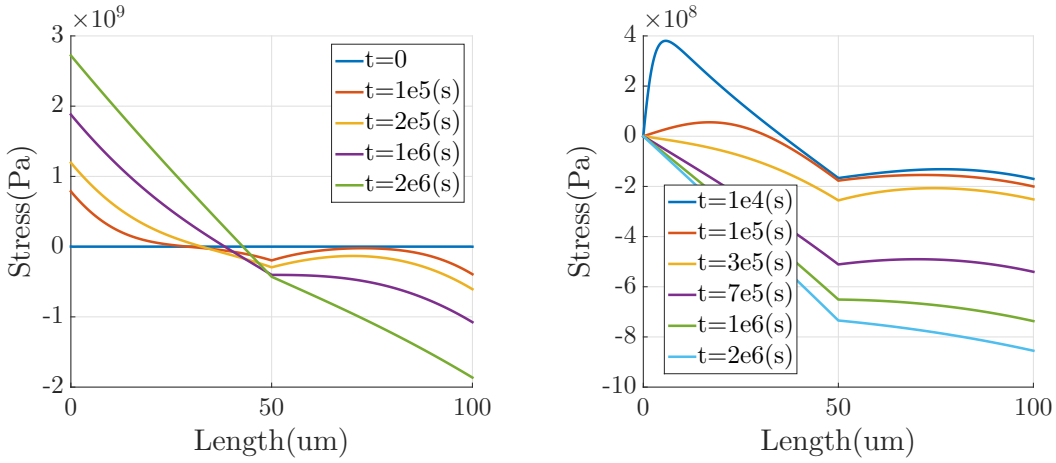


Figure 2.1: EM-stress evolution of a two segment wire in the (a)nucleation phase (b) incubation and growth phase.

recognized as critical volume, is not consumed by void. Once a void has nucleated when stress reaches a critical level at t_{nuc}, which is typically at or near a terminal node, the stress at the void boundary will immediately go to zero. However, the stress around the void will be close to the same stress level as immediately prior to the nucleation [6, 23]. As a result, a very large stress gradient will be formed around the voids at nucleation time, which can be described by [6, 24].

$$\nabla\sigma(t) = \frac{\sigma}{\delta}, \quad \text{on } \partial\Omega_L \cap \Gamma_{\text{void}}, \quad at \quad t_{nuc} < t < \infty \tag{2.5}$$

Γ_{void} denotes the void boundary, which can be a union of several discrete voids and δ is the effective thickness of the copper-void boundary [23]. The $\nabla\sigma(t)$ will serve as a force to move the void boundary toward the atomic flux direction. The void volume is determined by the stress distribution on the remaining section of the wire. Specifically, the void volume

$V_v(t)$ in a multi-segment wire will satisfy the following atom conservation equation [23, 25].

$$V_v(t) = -\int_{\Omega_L} \frac{\sigma(t)}{B} dV \tag{2.6}$$

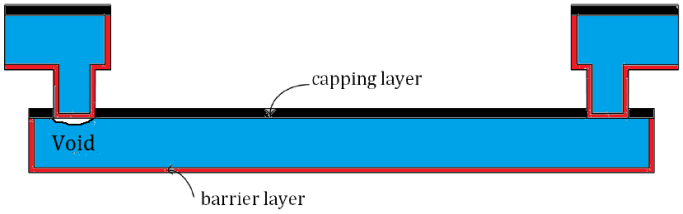
where Ω_L is the volume of the remaining interconnect wire and V is the volume of the wire and define as $V = W \times h \times l$, where W is width of wire and h is thickness of the wire and l is the length of the wire. Notice that Eq. (2.6) is also true for the void growth phase we discussed later. The incubation time can be defined as follows:

$$t_{nuc} < t < t_i \text{ where } V_v(t) < V_{crit} \tag{2.7}$$

where V_{crit} is the critical volume, which mainly depends on the cross section of via. In this work, we assume the diameter of via is equal to the width of wire W. So the critical volume can be written as $W \times h \times W$ since the resistance start to change when void's size reaches this volume. Fig 2.1(b) shows the stress evolution on a wire in the incubation phase. Stress at the cathode decrease to 0 very fast. Stress at anode keep decreasing until reach the steady state. This will lead to growth of the void since the integration of negative stress increase in Eq. (2.6). Void volume will saturate when the stress reach steady state.

Also, in the incubation phase, the void volume is not large enough to block the cross section of the via and smaller than critical volume. At the end of incubation phase a void will fill the cross section of the via. The void can cause either early failure or late failure of the wire [26]. Early failure typically happens in via above lines as shown in Fig. 2.2(a). The via will be blocked by the void and thus the connection to the upper layer will also be

blocked (capping layer is fabricated with dielectrics such as Si_3N_4 which does not conduct current flow). In this case, the wire will fail in the end of incubation phase. Late failure typically happens in via-below structures as shown in Fig. 2.2(b). When the void forms in a via-below line and reaches critical size, current can still go through the barrier layer and the resistance will increase in the growth phase.



(a)

(b)

Figure 2.2: Side-view of void formation: (a) void in a via-above line (early failure mode); (b) void in a via-below line (later failure mode).

2.1.3 Growth phase modeling

In the void growth phase, the void continues to grow. The major difference between growth phase and the incubation phase is that wire resistance starts to change (at least meaningfully). The reason is that current has to flow over the highly resistive barrier which is made with T_a/T_aN for instance, whose resistivity is much higher than C_u. Specifically, the wire resistance change can be approximately described as:

$$\Delta R(t) = \frac{V_v(t) - V_{crit}}{WH}\left[\frac{\rho_{Ta}}{h_{Ta}\,(2H+W)} - \frac{\rho_{Cu}}{HW}\right], at\ t_{inc} < t \tag{2.8}$$

where ρ_{Ta} and ρ_{Cu} are the resistivity of the barrier material (T_a/T_aN) and copper, W is the wire width, H is the copper thickness, and h_{Ta} is the barrier layer thickness.

2.2 Summary

This chapter presents a review of the three phase physics based EM model. Korhonen's equation based stress evolution in nucleation phase and post void condition in incubation phase and growth phase is presented. Besides, early failure and late failure condition is also demonstrated in this chapter.

Chapter 3

Fast Voltage Based Electromigration Immortality Analysis for Multi-Segment Copper Interconnect Wires

In this chapter, we present a novel and fast voltage based EM immortality check for general multi-segment interconnect wires. Instead of using current density as the key parameter, the new method estimates the EM-induced steady-state stress in general multi-segment copper interconnect wires based on a novel parameter, *Critical EM Voltage,* $V_{Crit,EM}$. The new VBEM method can naturally comprehend the impact of the topology of the wire structure on EM-induced stress.

3.1 Voltage-based EM stress estimation

In this section, we first present the compact EM model based on the solution of steady-state modeling of EM effects, which lead to the voltage-based assessment for EM failures for interconnect trees. We then show that the VBEM assessment is more suitable for multi-branch interconnect trees. We compare the results from the voltage-based VBEM with results from dynamic EM models to validate the new VBEM model on some complex interconnect wire structures.

3.1.1 Steady-state EM-induced stress modeling

At the steady state, the atomic flux becomes zero, which means that $\Gamma_\sigma = \Gamma_{EM}$. As a result, we have

$$\frac{\partial \sigma}{\partial x} = \frac{eZ\rho}{\Omega} j \tag{3.1}$$

On the other hand, the hydrostatic stress also leads to displacement of interconnect materials, which can be described as

$$\frac{\partial u}{\partial x} = \alpha\sigma \tag{3.2}$$

$$\sum_k u_{ik} w_{ik} = 0 \tag{3.3}$$

where, u_{ik} is the displacement and w_{ik} is the width at the k-th node on a branch between nodes i and k, and $M = 1/\alpha$ is the Young's modulus of the interconnect materials [27].

We note that Eq. (3.2) and (3.3) can be viewed as the atomic conservation (in the sense that the total number of metal atoms will remain the same during the migration

process in the wire) in the stress kinetics. We notice that a similar conservation equation was given in [7] for steady-state stress computation for multi-branch interconnects.

3.1.2 New voltage-based analysis for steady-state EM stress

Figure 3.1: A three-terminal wire, with the direction indicating electron flow.

One important observation is that the stress difference between two nodes at the steady state can be expressed in terms of voltage, instead of current density [27]. To demonstrate this, suppose that we have N nodes in the interconnect tree. We can discretize (3.1) in space, if we ignore the initial or residual stress, leading to

$$\sigma_k - \sigma_i = \frac{eZ\rho}{\Omega} l_{ik} j = \frac{eZ}{\Omega} V_{ik} \tag{3.4}$$

where σ_k is the steady-state stress at node k, and V_{ik} is the voltage or potential difference from node i and node k. When $V_{ik} > 0$, stress at node k is higher than the stress at node i (we assume that tensile stress is positive and compressive stress is negative).

From (3.2) and (3.3), we have

$$\sum_{k}^{N} a_k \sigma_k = 0 \tag{3.5}$$

where a_k is the total area of the branches connected to the node k. This equation represents the conservation in the stress kinetics. Note that (3.5) does not include the initial residual

stress as we will consider that later. For instance, in Fig. 3.1, the area for node 1 is $l_a w_a + l_b w_b$. For this case, (3.5) can be expressed as $l_a w_a \sigma_0 + (l_a w_a + l_b w_b)\sigma_1 + l_b w_b \sigma_2 = 0$. We also notice that for this case, we have

$$\sum_{k}^{N} a_k = l_a w_a + l_a w_a + l_b w_b + l_b w_b = 2(l_a w_a + l_b w_b) = 2A \tag{3.6}$$

where A is the total area of the branches in the wire. By considering (3.4), we can compute the stress for any node i in terms of all the node voltages with respect to the node i:

$$\sigma_i = \frac{\beta}{2A} \sum_{k \neq i}^{N} a_k V_k^i \tag{3.7}$$

where $\beta = \frac{eZ}{\Omega}$, and V_k^i is the nodal voltage at node k with respect to node i (node i is treated as the reference node).

If we select the reference node as the ground node with the lowest voltage of the segment, then the stress at this node can be expressed as

$$\sigma_g = \frac{\beta}{2A} \sum_{k \neq g}^{N} a_k V_k \tag{3.8}$$

where, V_k is the normal nodal voltage (with respect to cathode node g) at the node k in the wire. The cathode node is where the tensile stress σ_g is the highest in the segment.

If we further define a virtual voltage as

$$V_E = \frac{1}{2A} \sum_{k \neq g}^{N} a_k V_k \tag{3.9}$$

which can be viewed as **EM Voltage**, then according to (3.4), the stress at any node with nodal voltage V_i (with respect to ground node), can then be computed easily as in [27].

Furthermore, if the current density is not evenly distributed, an integration is required to calculate the *EM voltage* as follows:

$$V_E = \frac{1}{A} \iint V(x, y) dx dy \tag{3.10}$$

Then the steady-state stress at the node i can be calculated as:

$$\sigma_i = \beta(V_E - V_i) \tag{3.11}$$

This equation shows that the EM-induced stress at any node can be easily determined by the difference between the node voltage and the *EM voltage* for an interconnect segment. Therefore, EM stress determination is simplified to a problem of analyzing the node potentials and combining them with geometric information of the interconnect. Node potentials are readily available after circuit simulation and additional numerical computation is not required.

For EM effects, when the hydrostatic stress (EM-induced stress plus other existing stresses) at a node hits the critical stress, σ_{crit}, then the void starts to be nucleated and the resistance of the wire starts to increase over time, which can be measured experimentally [28, 29].

Assuming that initial or residual stress is σ_{init}, which is the same for all the nodes (this may not be true in general, but we have used the assumption to simplify our

presentation), then we can determine the voltage required for the nucleation to happen called **Critical EM Voltage,** $V_{Crit,EM}$ defined as below:

$$V_{crit,EM} = \frac{\Omega}{Ze}(\sigma_{crit} - \sigma_{init}) \tag{3.12}$$

The significance of $V_{Crit,EM}$ is that it essentially is the natural, but important extension of *Blech product* or *Blech limit* concept, which describes the EM immortality condition for a single-segment wire, to more general multi-segment interconnect wires (we will discuss this more in the next sub-section). Specifically, for any nodal voltage V_i, one only needs to further check whether

$$V_{Crit,EM} > V_E - V_i \tag{3.13}$$

If this condition is met for all the terminal voltages of the wire, then no EM failures will happen. Typically, for a multi-branch wire, when it is stressed only by positive voltages, if $V_E < V_{Crit,EM}$, no wire will have an EM failure because even the cathode node (with the lowest voltage) will not fail in this case. Otherwise, EM failures may happen. This makes the immortality check much more efficient as we typically only need to check one node for an interconnect segment, instead of checking every branch as is done for the traditional method. Note that if we have multiple nodes failing (3.13), all those nodes can lead to nucleation. However, for the question of EM immortality, as long as one node fails as per (3.13), the whole tree is mortal. On the other hand, if a wire fails the immortality check in (3.13), then more detailed and time-dependent stress analysis is required to determine the

time-to-failure or mean time-to-failure, which still remains a difficult problem for general multi-branch interconnect trees [30].

3.1.3 General and key VBEM equations

In this subsection, we summarize the general and key VBEM equations used for the EM immortality check. For a given arbitrary interconnect tree with N nodes, assuming the voltage in node i is V_i and the ground node is g and $V_g = 0$, then the stress at the node i can be computed as

$$\sigma_i = \frac{eZ}{\Omega}(\frac{1}{2A}\sum_{k\neq g}^{N} a_k V_k - V_i) \tag{3.14}$$

Given the *Critical EM Voltage,* $V_{Crit,EM}$, then EM immortality check for node i becomes

$$V_{Crit,EM} > \frac{1}{2A}\sum_{k\neq g}^{N} a_k V_k - V_i \tag{3.15}$$

We want to remark that as far as immortality/mortality is concerned, we are only interested in whether or not there exists at least one void formed in a given wire. If no void is formed, the wire is immortal, else the wire is mortal. Hence, we only need to look at the node with lowest voltage, the ground node or cathode node of the whole tree, as a result, (3.15) can be simplified to

$$V_{Crit,EM} > \frac{1}{2A}\sum_{k\neq g}^{N} a_k V_k \tag{3.16}$$

If (3.16) fails, then transient EM analysis will be carried out to find the void location and the nucleation time.

3.1.4 Relationship to the Blech limit

In this sub-section, we show how our VBEM analysis and **Critical EM Voltage,** $V_{Crit,EM}$ is related to the existing Blech limit. We first show that the Blech product essentially is the voltage-based EM assessment for just one wire segment. The proposed VBEM method can actually be viewed as the general extension of this technique to multi-branch interconnect wires.

Specifically, let L be the length of a single wire and j the current density of the wire. Starting with the steady-state condition of EM stress shown in (3.1), which is also called the *Blech condition*, if we integrate (3.1) along the line, we obtain

$$\sigma(x) = \sigma_{init} + \frac{eZ\rho j}{\Omega}x \tag{3.17}$$

where, σ_{init} is the residual stress. The maximum tensile stress can be achieved at the cathode end of the wire ($x = L$).

If the critical stress that the wire can withstand is σ_{crit}, we can define the critical product for EM failure as

$$(jL)_{crit} = \frac{\Omega(\sigma_{crit} - \sigma_{init})}{eZ\rho} \tag{3.18}$$

which is called the *Blech limit* or *Blech product* [5]. A wire is immortal for EM if it satisfies $jL < (jL)_{crit}$. As a result, the Blech product can help identify all the immortal wires efficiently.

Notice that if ρ is the resistivity, ρLj is actually the voltage across the wire, then (3.18) becomes

$$(jL)_{crit} * \rho = V^b_{crit} = \frac{\Omega(\sigma_{crit} - \sigma_{init})}{eZ}. \tag{3.19}$$

where V^b_{crit} is actually the critical voltage for this single wire. As we can see, this equation is the same as (3.12) for the single wire case. However, we want to stress that the new **Critical EM Voltage,** $V_{Crit,EM}$ is more general than the Blech limit due to the following reasons: first, the failure criterion is no longer associated with current density and length of a specific wire segment. In other words, the existing Blech product, jL, does not work any more in this case as the jL of individual branch not only depends on the critical stress, but also depends on the wire structures as the stresses in each wire segment are not independent as they affect each other. However, the proposed $V_{Crit,EM}$ concept can be applied to general interconnect trees with multi-segment wires as one single EM immortality criterion. Second, $V_{Crit,EM}$ still retains the benefits of the Blech limit as the voltage values can be measured directly based on the pass/fail determination of wires from experiments. Third, it agrees with the Blech limit for the single-wire segment case, which essentially validates the proposed method. It has the potential be used as a new design rule parameter between the foundry and design teams to replace or extend the current Blech limit parameter.

3.1.5 Steady-state analysis

In this sub-section, we first show that the proposed VBEM analysis agrees with the results from the steady-state results computed from the finite difference method. Then we show that it agrees with a published closed-form expression for a special case. We demonstrate this using one example, a three-terminal wire as depicted in Fig. 3.1.

We first compare with the finite difference method. Let the total length be L. The segment lengths l_a and l_b in the three-terminal wire are equal to $\frac{L}{2}$. We then use this wire segment length as the spatial step size and use the backward difference method for boundary derivation as shown in [31]. We remark that the matrix derived by the 1D finite difference method actually is singular. The reason is that the atomic conservation in the stress kinetics is not observed in this case. In order to resolve this problem, Eq. (3.5) is introduced to replace one row in the matrix. The resulting system of the equations for the three-terminal wire case is presented as below:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\sigma_0} \\ \dot{\sigma_1} \\ \dot{\sigma_2} \end{bmatrix} = \frac{\kappa}{(L/2)^2} \begin{bmatrix} -1 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & -1 \end{bmatrix} \times \begin{bmatrix} \sigma_0 \\ \sigma_1 \\ \sigma_2 \end{bmatrix} + \begin{bmatrix} \frac{2\kappa G_a}{L} \\ 0 \\ -\frac{2\kappa G_b}{L} \end{bmatrix} \tag{3.20}$$

Here, G_a and G_b are the EM driving forces corresponding to segments a and b, respectively. We then rewrite these equations into the following format:

$$\begin{aligned} \mathbf{C}\dot{\sigma}(t) &= \mathbf{A}\sigma(t) + \mathbf{B} \\ y(t) &= \mathbf{E}\sigma(t) \end{aligned} \tag{3.21}$$

In Eq. (3.21), $\mathbf{E} = (1, 0, 0)$, which means we are looking at the steady state of node 0 which

is the cathode node. Then, a Laplace transform can be applied and the transfer function can be obtained as:

$$F(s) = \mathbf{E}(s\mathbf{C} - \mathbf{A})^{-1}\mathbf{B} \tag{3.22}$$

Under step input, which is $\frac{1}{s}$ in frequency domain, the Final Value Theorem can be used to obtain the stress at steady state. If $\lim_{t\to\infty} f(t)$ has a finite limit under a step input, the Final Value Theorem for a function under the step input can be expressed as:

$$\lim_{t\to\infty} f(t) = \lim_{s\to 0} sF(s)\frac{1}{s} = F(s)|_{s=0} = -\mathbf{E}\mathbf{A}^{-1}\mathbf{B} \tag{3.23}$$

We notice that $(-A)^{-1}$ is

$$(-A)^{-1} = \frac{L^2}{4\kappa} \times \frac{1}{4} \begin{bmatrix} 3 & -1 & -1 \\ -1 & -1 & -1 \\ -1 & -1 & 3 \end{bmatrix} \tag{3.24}$$

Then, we can obtain the steady-state result of the system:

$$\begin{aligned} \sigma_{steady} = F(s)|_{s=0} &= (\frac{3}{4} \times \frac{2\kappa G_a}{L} + \frac{1}{4} \times \frac{2\kappa G_b}{L})\frac{L^2}{4\kappa} \\ &= \frac{(3G_a + G_b)L}{8} \end{aligned} \tag{3.25}$$

On the other hand, we can compute steady-state stress at the cathode node (node

0) based on the voltage-based method as:

$$\begin{aligned}\sigma_{steady} = V_E \times \beta = \frac{A_1V_1 + A_2V_2}{2A} \times \beta \\ = \frac{j_aL\rho + j_aL\rho/2 + j_bL\rho/2}{4} \times \beta = \frac{(3G_a + G_b)L}{8}\end{aligned} \tag{3.26}$$

where $V_E = \frac{A_1V_1+A_2V_2}{2A}$ is the *EM voltage* at the cathode node and $\beta = \frac{eZ}{\Omega}$. As we can see, the results from the two methods are identical. This example gives another theoretical validation of the proposed VBEM analysis method, which agrees with the steady-state results from the finite difference method.

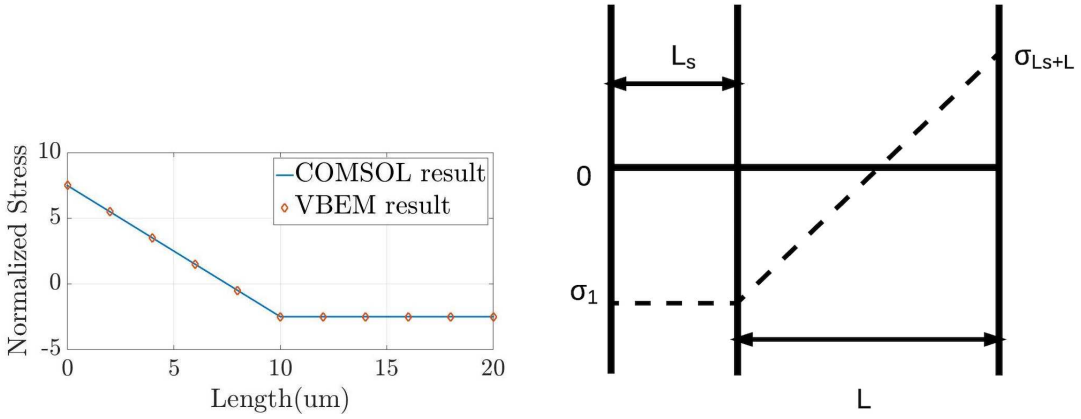


Figure 3.2: (a) A 3-terminal wire with inactive (passive) sink, with the cathode at node 2. (b) The steady-state stress distribution of a 3-terminal wire with inactive (passive) sink.

Now we show that the discretization schemes (by using different discretization sizes) will not change the steady-state results. Fig. 3.3 shows stress analysis using the simplified FDM method comparing COMSOL result of Korhonen's equation. It can be observed that the steady-state results under different discretization sizes are same although their transient behaviors are different. In other words, discretization errors will not affect the final steady-state results from FDM.

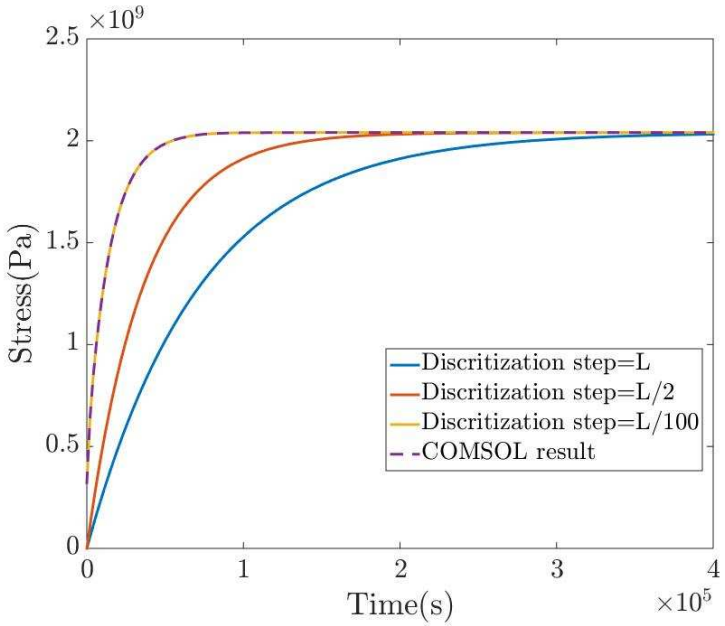


Figure 3.3: Different discretization steps (sizes) lead to the same steady-state stress.

In addition to analyzing the FDM steady state results considering different discretization schemes, we further show some pole information from the FDM analysis of Korhonen's equation. We use the same 3-terminal wire shown in Fig.3.1 as an example. In this case, we discretize the two segments into 21 nodes (so we have 21 poles) instead of just the 3 boundary nodes presented in (3.20). As we can see from Table 3.1, all the poles are real poles and the EM system is stable. The EM-induced stress basically is progressing monotonically for a given step current input. Such a monotonic nature of EM-induced stress is important to ensure the steady state is sufficient for the EM immortality check. Our study shows that this is the case for all the examples we analyzed. We also remark that although there is still a possibility of overshoots for a system with negative poles, for our case, it is rarely observed since they are only theoretically possible. Practically, it does not seem necessary to worry about the oscillating behaviors in the stress evolution process.

Table 3.1: The computed poles for the 3-terminal wire

Poles values ($\times 10^3$)				
-1.7541	-1.7248	-1.6767	-1.6107	-1.5286
-1.4319	-1.3230	-1.2042	-1.0783	-0.9479
-0.8161	-0.6857	-0.5598	-0.4410	-0.3321
-0.2354	-0.1533	-0.0873	-0.0392	-0.0099
-0.0000	–	–	–	–

Furthermore, we notice that recent work in [32] gives the closed-form expression (which is consistent with the measured experimental results) of steady-state stress for a special three-terminal wire case with an inactive (passive) segment (Fig. 3.2(a)), and the stress profile (Fig. 3.2(b)). In this case, the segment L_s has zero current (so it is inactive or passive), while the active segment L has current flow j. The cathode is located at node 2. It was shown in [32] that the steady-state stress at any location x in the active segment $\sigma(x)$ is given by:

$$\sigma(x) = \frac{e\rho Zj(x - L_s)}{\Omega} + \sigma_1 \tag{3.27}$$

where σ_1 is the stress in the inactive sink as shown in Fig. 3.2(b), which is given by

$$\sigma_1 = -\frac{e\rho ZjL^2}{2\Omega(L + L_s)} \tag{3.28}$$

As a result, the stress at the cathode node ($x = L_s + L$) will become

$$\begin{aligned} \sigma(L + L_s) &= \frac{e\rho ZjL}{\Omega} - \frac{e\rho ZjL^2}{2\Omega(L + L_s)} \\ &= \frac{e\rho Zj}{\Omega}\left[L - \frac{L^2}{2(L + L_s)}\right] \end{aligned} \tag{3.29}$$

On the other hand, based on the VBEM method, we have $V_0 = V_1 = jL\rho$ (as there is

no current in L_s and $A_0 = L_s w$ and $A_1 = Lw + L_s w$, where w is the width of the two segments (assuming two segments have the same width). Then, the stress at the cathode node, $\sigma(L + L_s)$, can be computed by

$$\begin{aligned}\sigma(L+L_s) &= \beta \times V_E(L+L_s) = \beta \times \frac{A_0 V_0 + A_1 V_1}{2A} \\ &= \beta \times \left[\frac{jL\rho(L_s w + Lw) + jL\rho L_s w}{2(Lw + L_s w)}\right] \\ &= \beta \times \left[\frac{jL\rho}{2} + \frac{jL\rho L_s}{2(L+L_s)}\right] \\ &= \frac{e\rho Z j}{\Omega}\left[L - \frac{L^2}{2(L+L_s)}\right]\end{aligned} \tag{3.30}$$

Comparing (3.29) and (3.30), we can see again that the VBEM method agrees exactly with the closed-form expression for this particular case given by [32].

3.1.6 Study of some special cases

In this sub-section, we study three multi-branch interconnect structures to illustrate the proposed method. The three structures consist of a straight-line 3-terminal wire in Fig. 3.4, a T-shaped 4-terminal wire in Fig. 3.7, and a comb structure wire in Fig. 3.9. We stress that the proposed method can be applied to any multi-branch tree-structured interconnects.

The straight-line 3-terminal wire

Figure 3.4: Interconnect examples for EM analysis for a straight-line 3-terminal wire.

The straight-line 3-terminal wire is shown in Fig. 3.4. In this wire, the node 0 is treated as the ground node. Note that current densities in the two segments are j_a and j_b, which are determined by the rest of the circuit and may not be the same. Then, the EM stress equation becomes:

$$
\begin{array}{lll}
V_0 = 0, & A_0 = l_a w_a, & \sigma_0 = \beta V_E \\
V_1 = j_a l_a \rho, & A_1 = l_a w_a + l_b w_b, & \sigma_1 = \beta(V_E - V_1) \\
V_2 = j_b l_b \rho + j_a l_a \rho, & A_2 = l_b w_b, & \sigma_2 = \beta(V_E - V_2)
\end{array}
$$

$$A = A_0 + A_1 + A_2$$

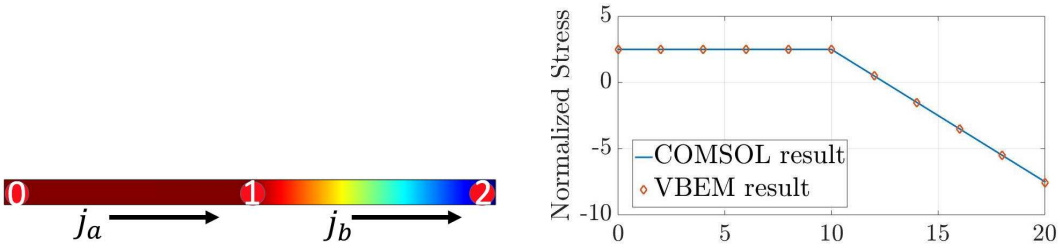
where

$$V_E = \frac{V_0 A_0 + V_1 A_1 + V_2 A_2}{2A} = \frac{V_1 A_1 + V_2 A_2}{2A}$$

where $\beta = \frac{eZ}{\Omega}$.

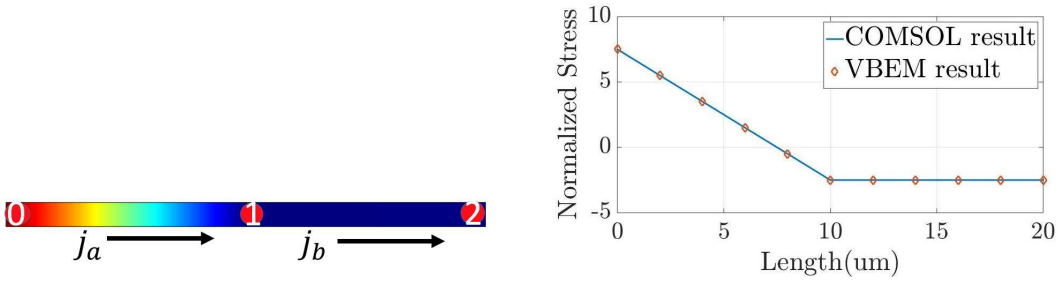
Passive sink and passive reservoir configurations, described in [32], are typical elements in the general interconnect tree. They are used as test cases in this sub-section. Fig. 3.5 and Fig. 3.6 show the steady-state stress for the cases with passive reservoir (segment a with $j_a = 0$) and passive sink (segment b with $j_b = 0$). Here we define "passive" and "active" as representing zero current density and non-zero current density, respectively. The analysis will focus on mitigating the EM effect in the active segment. It can be observed from

Fig. 3.5 that the passive reservoir (segment a) is characterized with higher stress compared with the active sink (segment b). Thus, the void will first nucleate in the reservoir which can relax the EM effect in the sink. On the contrary, as shown in Fig. 3.6, the existence of passive sink (segment b) will lead to higher stress in the active reservoir (segment a), which accelerates the void formation thus leading to EM failure in the reservoir. A comparison of the steady-state stress predicted by VBEM with the finite element analysis (COMSOL) simulation in both cases has demonstrated an excellent agreement.



(a) (b)

Figure 3.5: (a) 2-D stress distribution on wire at steady state for passive reservoir (b) EM stress versus length at steady state.



(a) (b)

Figure 3.6: (a) 2-D stress distribution on wire at steady state for passive sink (b) EM stress versus length at steady state.

T-shaped 4-terminal wire with stub

The structure of the T-shaped 4-terminal wire is shown in Fig. 3.7. In this case, we have three segments which connect through the middle node 1. Current densities are j_a, j_b, and j_c on the three branches. In this case, if we make the branch c (the vertical branch), the stub (its current density is set to zero, $j_c = 0$), then the EM stress can also be obtained:

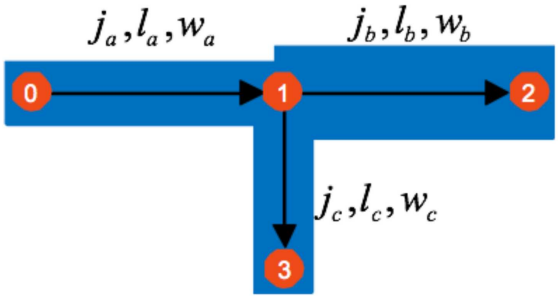


Figure 3.7: Interconnect examples for EM analysis for T-shaped 4-terminal wire.

$$V_0 = 0, \qquad A_0 = l_a w_a$$

$$V_1 = j_a l_a \rho, \qquad A_1 = l_a w_a + l_b w_b + l_c w_c$$

$$V_2 = j_b l_b \rho + j_a l_a \rho, \qquad A_2 = l_b w_b$$

$$V_3 = j_a l_a \rho + j_c l_c \rho, \qquad A_3 = l_c w_c$$

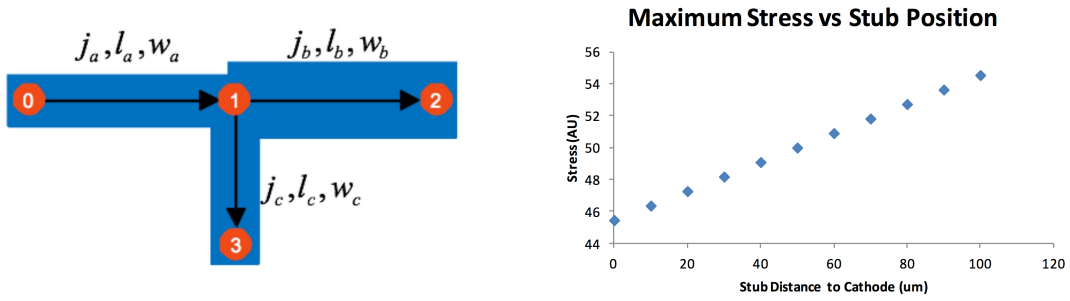
$$\sigma_0 = \beta V_E, \qquad \sigma_1 = \beta(V_E - V_1)$$

$$\sigma_2 = \beta(V_E - V_2), \qquad \sigma_3 = \beta(V_E - V_3)$$

$$A = A_0 + A_1 + A_2 + A_3$$

where

$$V_E = \frac{V_1 A_1 + V_2 A_2 + V_3 A_3}{2A}$$



(a) (b)

Figure 3.8: (a) Illustration of T-junction interconnect with directed graph inserted to indicate electron flow. (b) Stress at the cathode end increases linearly as the stub is placed further away from the cathode end.

The stub acts as a sink when it is close to an anode and serves as a source when it is close to a cathode. The distance between the stub and the cathode and the length of the stub can be important factors related to EM stress. As shown in Fig. 3.8(b), when the stub is moved away from the cathode node (node 0), it allows more atoms to migrate to the stub and thus creates more tensile stress at the cathode node. This is a commonly seen structure in interconnect circuit design. We will discuss the effects of the distance between the stub and the cathode as well as the length of the stub on EM stress. Thus, by adjusting the stub location and length, we can adjust the stress at the cathode node to fix potential EM failures in the physical design.

Interconnect wires with comb structure

Now, we study a more complicated interconnect structure, which is the comb or ladder structure as shown in Fig. 3.9.

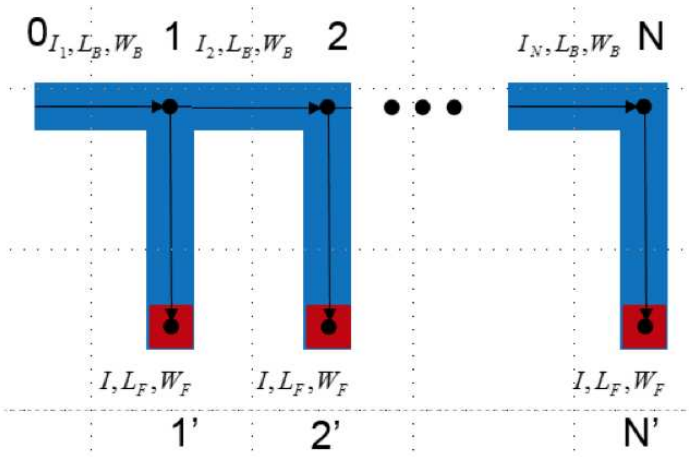


Figure 3.9: Comb structure interconnect examples for EM analysis.

In this comb-structured interconnect, we have N fingers, in which each finger structure is assumed to be the same. R_{sh} is the sheet resistance of the metal and I is the current along each finger. We assume that node 0 is still the ground node. L_B and W_B are the length and width, respectively, for the body structures, L_F and W_F are the length and width, respectively, for the fingers. i refers to i-th node on the body and i' is the node on the i-th finger.

The total area connected to node k, except node N, is A_k and total area connected to node k' is $A_k{}'$, total area connected to node N is A_N, and the total area of the whole

comb structures, A, can be expressed as:

$$A_k = 2W_B L_B + W_F L_F, \qquad A_k{}' = W_F L_F,$$
$$A_N = W_B L_B + W_F L_F,$$
$$A = N(W_B L_B + W_F L_F)$$

Note that since the N-th node is only connected to one part of the body structure, the total area connected with node N is different from other nodes. Current flows in the same direction (which is opposite to the arrows in the figure), the highest EM tensile stress will be generated at node 0 because it has the lowest potential. Hence, in this case, we only need to check V_E against the critical potential.

The potential at each node for V_k and $V_k{}'$ can be obtained as:

$$V_k = \left[Nk - \frac{k(k-1)}{2}\right] \times I \times \frac{L_B}{W_B} R_{sh},$$
$$V_k{}' = V_k + I \times \frac{L_F}{W_F} R_{sh}$$

Finally, the EM stress of the comb structure can be obtained as:

$$V_E = \frac{IR_{sh}}{12} \times \left[\frac{(N+1)(4N-1){L_B}^2}{W_B L_B + W_F L_F} + \frac{2(N+1)(2N+1)L_B L_F W_F / W_B + 6{L_F}^2}{W_B L_B + W_F L_F}\right], \tag{3.31}$$
$$\sigma_0 = \beta V_E$$

As shown in the above equation, three factors, N, L_F and L_B have a strong influence on stress. Fig. 3.10 shows how the EM-induced stresses at the node 0 change with L_F and L_B. As we can see, L_B has a much larger impact on the stress than L_F, which can be used for EM optimization. Both L_F and L_B have nonlinear impacts on the stress and this nonlinear trend is more clear for L_F

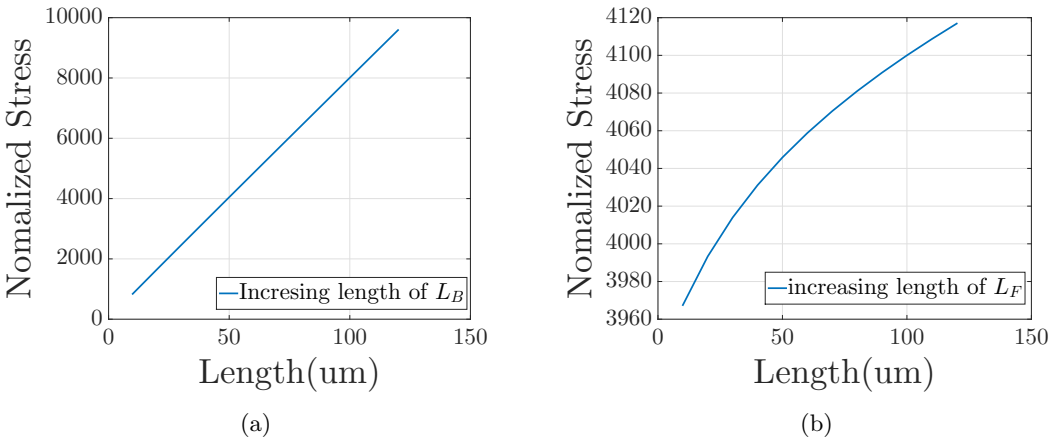


Figure 3.10: (a) EM stress validation for each comb structure interconnect with changing L_B. (b) EM stress validation for each comb structure interconnect with changing L_F.

Other trends of stress change will be analyzed and discussed in the numerical results section.

3.1.7 Current crowding impact analysis

In this sub-section, we study the impact of the current density distributions on the proposed VBEM analysis method. The VBEM method assumes that current density is evenly distributed. However, this is not always the case since for real interconnect wires, the current density may vary and become larger around the corner areas (the current crowding effect). We observe that if the width is not much smaller than the segment length, the

current crowding effect can be quite significant. In this case, the calculated nodal voltage will be less accurate for stress calculations.

To consider the current crowding effects, instead of using (3.9) to compute the **EM voltage**, we need to perform the area integration of voltage using (3.10) to compute the final steady-state stress after current and voltage distributions are computed. As we will show in this subsection, this will lead to more accurate results compared to the results using (3.9), and that the proposed method can be extended to consider current crowding effects. The area used for the integration is the total areas of the wires in 2D case (although we show 3D structures of the wires) since (3.10) is for 2D integration. In principle, the integration can be done over the 3D volume. It does not have to be restricted to 2D integration. We also remark that the VBEM method will be more expensive for the numerical 2D integration operation. This is due to the nature of the current crowding modeling problem. One has to compute the detailed current and voltage distributions first using expensive numerical methods such as the finite element method, to account for the effects of current crowding on EM risks.

In the following, we study two wire structures to assess the impact of current crowding effects. In Fig. 3.11, an L-shaped wire with 3 nodes is used as the first test structure. Here, l_a and l_b are not much larger than the width of wire w. A voltage is applied on node 1 and node 0 is the ground node. Node 2 is connected with node 1 through a stub with a current density of 0.

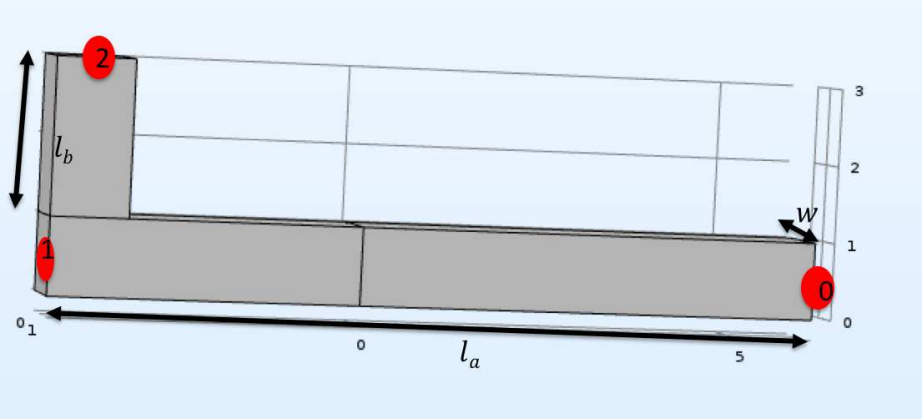


Figure 3.11: L-shaped wire structure with 3 nodes.

In this case, the current density distribution around node 1 is not uniform, as shown in Fig. 3.12. COMSOL simulation is used to obtain that non-uniform current density distribution as well as voltage distribution. At node 1, the current density is smaller than the current density on the other part of the branch. This means that if the nodal voltage is used to calculate the stress, it will be smaller than the actual condition. On the other hand, if the branch is longer (compared to the width of the wire), the current crowding has a smaller effect on the final steady-state stress.

Table. 3.2 summarizes the results for two cases for the stress values at the cathode node. In case 1, $l_a = 10$, $l_b = 2$, and $j = 3MA/cm^2$ with voltage difference 0.03V between node 0 and 1, In case 2, $l_a = 4$, $l_b = 2$, and $j = 3MA/cm^2$ with voltage difference is 0.012V between node 0 and 1. Column *Crowding* indicates that the current-crowding effect is considered by using (3.10) and *Err1* is the relative error of considering the crowding effect compared to COMSOL. Column *noCrowding* indicates that no current crowding effect is computed using (3.9) and *Err2* is the error without considering the crowding effect.

As we can see from Table. 3.2, for the long branch (case 1), the error for the

method without considering current crowding is 1.53% and the error increases to 7.72% for the shorter branch (case 2), which is quite significant. However, if the current crowding effect is considered (as shown in *Crowding* column), the errors become smaller (less than 2%).

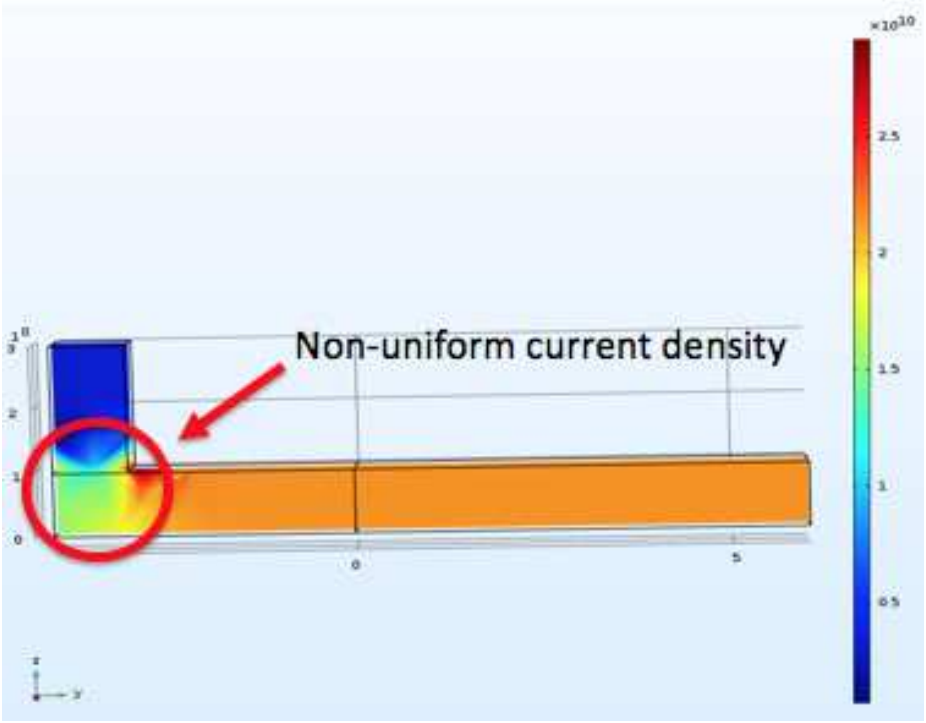


Figure 3.12: L-shaped wire structure with current-crowding effects.

The second structure we study is the U-shaped wire shown in Fig. 3.13. Its segment length is not significantly larger than the width in this case and current density distribution can be seen in Fig. 3.14, with the current-crowding effect being very visible at nodes 2 and 3.

Table 3.2: EM stress calculated with and without current-crowding effects

Cases	COMSOL	Crowding	Err1	NoCrowding	Err2
1	286MPa	284MPa	0.94%	282MPa	1.53%
2	97.7MPa	96.1MPa	1.60%	90.4MPa	7.72%

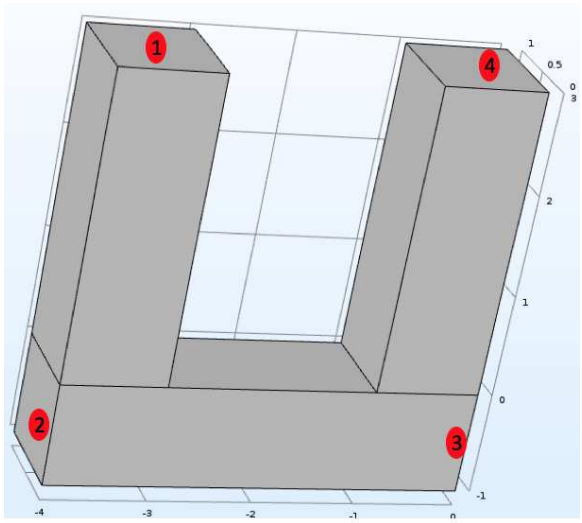


Figure 3.13: U-shaped wire structure with 4 nodes.

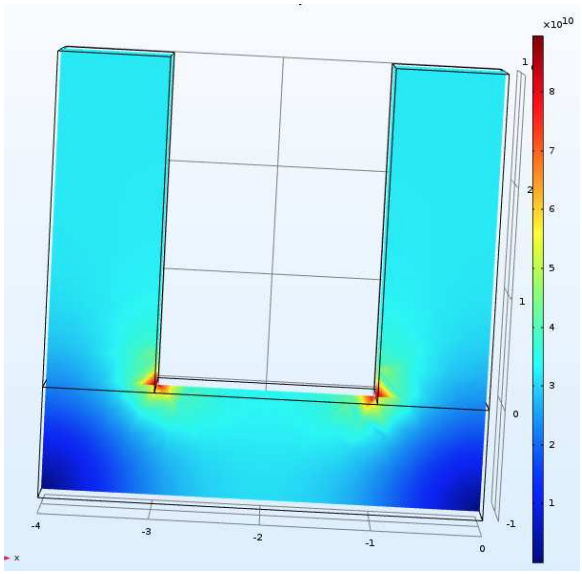


Figure 3.14: U-shaped wire structure with current-crowding effects.

Table 3.3: Stress values at each node for the U-shaped structure

Cases/Nodes	1	2	3	4
COMSOL	345.5Mpa	104.2Mpa	-104.2Mpa	-345.5Mpa
Crowding	339.0MPa	102.2MPa	-102.2MPa	-339.0Mpa
Error	1.74%	1.95%	1.95%	1.74%
No Crowding	339Mpa	135.6Mpa	-135.6Mpa	-339Mpa
Error	1.74%	23.16%	23.16%	1.74%

For the U-shaped structure, we apply 0.05V to node 1 and 0V voltage to node 4. The result is shown in Table 3.3 for the stress values at the four nodes. Row *Crowding* indicates that the current-crowding effect is considered and row *noCrowding* indicates that the crowding effect is not considered.

As we can see, for nodes 1 and 4, even when current crowding is not considered, the error is only 1.74%, which is close to the errors of cases considering current crowding. However, for nodes 2 and 3 if current crowding is not considered, the errors increase to 23.16%. On the other hand, the error reduces to 1.95% when current crowding is considered. Thus, in both cases, if the wire segment is long enough (more than 10 times of wire width), the current-crowding impact on stress values by the VBEM method is not significant.

3.1.8 Application to mesh-structured interconnect wires

In this section, we study if the proposed VBEM analysis method can be applied to mesh-structured interconnect wires, which can be used at the cell-level layout design and can be vulnerable to EM failure as well.

Fig. 3.15 shows a 4×4 mesh structure with 16 nodes. In order to calculate the V_E, nodal voltages and areas connected with each node are required. A voltage is applied on node 4 and node 13 is set to be ground node. The voltage of each node can be measured or analyzed by SPICE. Areas connected to each node are different at different locations. For the nodes at corners (1,4,13,16), the area connected with them is $2WL$, where W is the width of the wire and L is the distance between adjacent nodes.

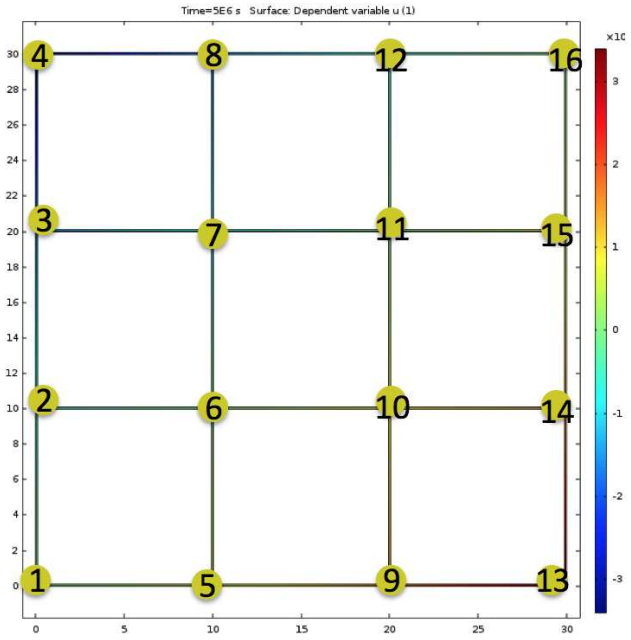


Figure 3.15: A 4×4 mesh-structured wire.

For the nodes at the boundaries (2,3,5,8,9,12,14,15), the area connected with them is $3WL$. For the nodes in the middle (6,7,10,11), the area connected with them is $4WL$. The total area is the area summation of each branch, which is $48WL$. Then the V_E can be obtained as:

$$
\begin{aligned}
V_E = (&2WL(V_1 + V_4 + V_13 + V_16) \\
&+ 3WL(V_2 + V_3 + V_5 + V_8 + V_9 + V_12 + V_14 + V_15) \\
&+ 4WL(V_6 + V_7 + V_10 + V_11))/48WL \\
\sigma_{max} = \sigma_{13} &= \beta V_E
\end{aligned}
$$

In our stressing set up, we apply 0.005V to node 4 and 0V (ground) to node 13. The length of each branch is 10um and width is 0.1um. Table 3.4 shows the result of the test case. As we can see, for all the cases, the VBEM method leads to less than 0.17% error compared to COMSOL, which shows that the proposed VBEM method can be directly applied to mesh-structured wires.

3.1.9 Application to IBM power grids

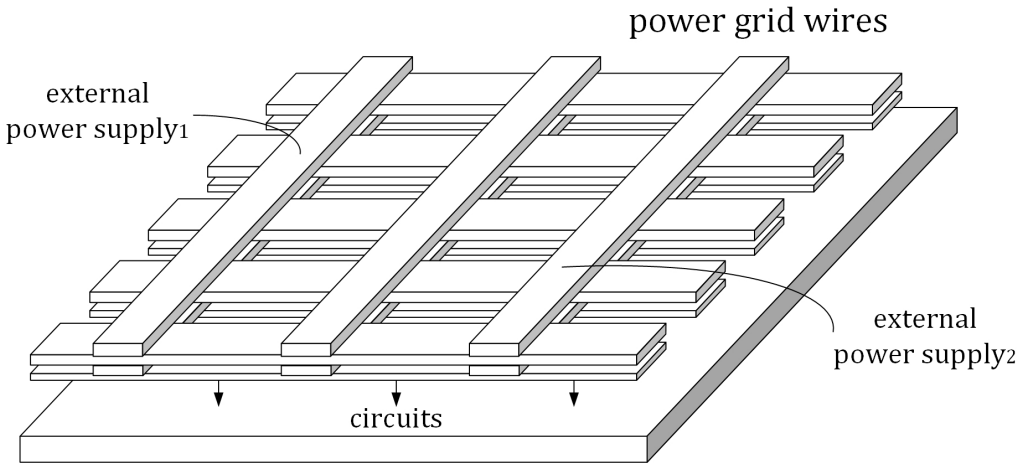


Figure 3.16: A small portion of a typical power supply network [2].

Besides small interconnect structures, we also validate the proposed method on a large practical IBM power grid benchmark [2]. A portion of the power grid network is shown in Fig. 3.16. Details of the benchmark are shown in Table 3.5. In this experiment, critical stress is $500MPa$ [33]. The critical voltage is 3.694×10^{-3} V. For the IBM power grid networks, COMSOL based FEM analysis method is too slow. Instead, we use a recently proposed eigen-function-based stress analysis method for multi-segment interconnect trees [34] as the baseline for comparison. The proposed VBEM and baseline methods were both

Table 3.4: Stress condition for mesh structure

Nodes	1	2	3	4
Voltage	2.498e-3V	2.880e-3V	3.656e-3V	5e-3V
COMSOL	0.2441Mpa	-51.61MPa	-157.0Mpa	-339.5MPa
VBEM	0.2439Mpa	-51.53Mpa	-156.8Mpa	52.07Mpa
Error	0.0771%	0.1549%	0.1572%	0.1609%
Nodes	5	6	7	8
Voltage	2.116e-3V	2.498e-3V	3.079e-3V	3.652e-3V
COMSOL	52.15Mpa	0.2118MPa	-78.62Mpa	-156.6MPa
VBEM	52.07Mpa	0.2116Mpa	-78.50Mpa	-156.3Mpa
Error	0.1543%	0.095%	0.1547%	0.1572%
Nodes	9	10	11	12
Voltage	1.347e-3V	1.921e-3V	2.502e-3V	2.884e-3V
COMSOL	156.8Mpa	78.62MPa	-0.2114Mpa	-52.15MPa
VBEM	156.3Mpa	78.50Mpa	-2.110Mpa	-52.07Mpa
Error	0.1571%	0.1545%	0.1529%	0.1545%
Nodes	13	14	15	16
Voltage	0	1.344e-3V	2.120e-3V	0.250e-3V
COMSOL	338.5Mpa	157.0MPa	51.61Mpa	-0.2441MPa
VBEM	339.8Mpa	156.8Mpa	51.53Mpa	-0.2438Mpa
Error	0.1543%	0.095%	0.1547%	0.1572%

implemented with $C/C++$ for comparison. The comparison works were carried out on a workstation with 2 Intel Xeon E5-2698 CPUs and 128GB memory.

We can see thatthe VBEM method has significant acceleration compared to the baseline method. For *ibmpg1*, VBEM takes only 0.69 seconds to simulate all 689 trees, which translates to a 1319X speed-up over the baseline method. All the trees in *ibmpg1* are small trees with the maximum number of branches in a tree being 30. With larger trees, the acceleration rate decreases, but one still sees 30.86X acceleration for *ibmpg2*, 4.43X acceleration for *ibmpg3*, and 4.91X acceleration for *ibmpg4*. Note that the lengths

Table 3.5: property of IBM benchmarks

Name	ibmpg1	ibmpg2	ibmpg3	ibmpg4
#node	11572	61797	407279	474836
#branch	5580	61143	399201	384709
#max branch	30	192	965	571
#trees	689	462	7388	9358
#failed trees	249	91	1585	0
VBEM(s)	0.69	29.63	3999.97	4565.27
Baseline(s)	910.56	973.65	17720.97	22456.13
Acceleration	1319X	30.86X	4.43X	4.91X

of branches are not the same. As can be seen in *ibmpg4*, although it has trees with many branches, the lengths of branches are small so all trees are immortal in this case.

We note that the baseline method is a transient EM analysis while the VBEM method is a steady-state method. Another saving brought by the VBEM method is the percentage of the immortal tree count over the total tree count, as the immortal trees will not need the costly transient EM analysis. The immortal tree number can be significant compared to the total tree count (ranging from 63.8% in *ibmpg1* to 100% in *ibmpg4* case. Hence, the percentage of savings from the proposed VBEM method is problem-specific and can be very significant.

3.2 Numerical validation results and discussions

In this section, we validate the proposed voltage-based EM (VBEM) check tool against numerical analysis results. We validate the VBEM method against the results by a finite element analysis (FEA) tool, COMSOL [35], based on the dynamic stress evolution described by Korhonen's equation. In the following, we list the results for the three structures we have discussed.

3.2.1 Results for straight-line 3-terminal interconnects

The parameters used for the validation cases are summarized in Table 3.6 and the results for the 3-terminal wire are shown in Fig. 3.17, which shows the largest tensile stress at node 0. We compare our results against COMSOL and another published EM numerical simulator, XSim [36], which has been validated by measured results [36, 37].

Table 3.6: Parameters for each straight-line 3-terminal interconnect case

Case	**Branch a**			**Branch b**		
	l	w	j	l	w	j
	μm	μm	MA/cm^2	μm	μm	MA/cm^2
1	25	1	1.25	0	1	0
2	25	1	1.25	175	1	0
3	25	1	1.25	175	1	0.125
4	25	1	1.25	175	1	0.625
5	25	1	1.25	175	1	1.25
6	10	0.1	10	25	1.25	1.25
7	10	0.2	5	25	1.25	1.25
8	10	0.3	3.3	25	1.25	1.25
9	10	0.4	2.5	25	1.25	1.25
10	10	0.5	2	25	1.25	1.25

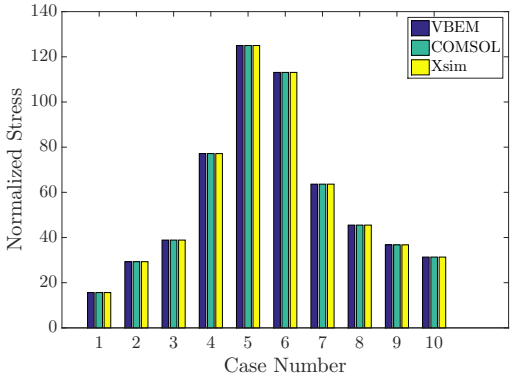


Figure 3.17: Steady-state EM stress comparisons for each straight-line 3-terminal interconnect case (x: case number, y: EM stress at the node 0 (cathode node)).

As demonstrated in Fig. 3.17, the results of the proposed method agree well with COMSOL results. To further validate the new method, we compared our results against the EM simulator XSim for steady-state results. As we can see, the VBEM method also agrees very well with XSim, which further validates the proposed method. With the increase in length and current density in the branch b, the EM stress increases. If the current density in branch a decreases, EM stress also decreases.

3.2.2 Results for T-shaped 4-terminal interconnect

We next validate the VBEM method on the T-shaped 4-terminal interconnect case. Again, we list the parameters used for the validation cases in Table. 3.7 and results of 3-terminal wires are shown in Fig. 3.18, which shows the largest tensile stress at the node 0.

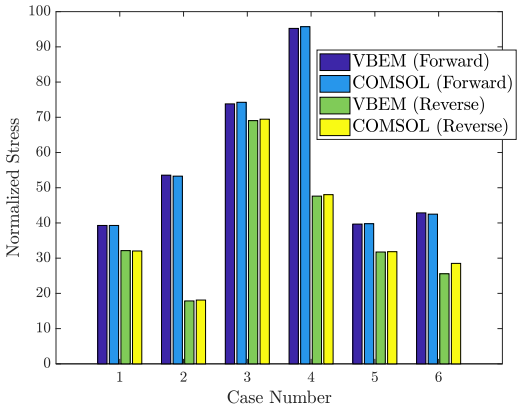


Figure 3.18: EM stress validations for each T-shaped 4-terminal interconnect case (x: case number, y: EM stress at the node 0 (cathode node)).

During the analysis of the T-shaped interconnect, forward and reverse currents

Table 3.7: Parameters for each T-shaped 4-terminal interconnect case ($l = \mu m$, $w = \mu m$, and $j = MA/cm^2$)

Branch	**Case**	1	**2**	**3**	**4**	**5**	**6**
a	**l**	6	10	11	20	20	20
	w	0.14	0.14	0.14	0.14	0.28	0.28
	j	7.142	7.142	7.142	7.142	3.571	3.571
b	**l**	4	0	9	0	0	0
	w	0.14	0.14	0.14	0.14	0.28	0.28
	j	7.142	7.142	7.142	7.142	3.571	3.571
c	**l**	5	5	10	10	5	10
	w	0.28	0.28	0.14	0.14	0.14	0.14
	j	0	0	0	0	0	0

Table 3.8: EM stress validations for comb structure interconnect cases.

Comb Case	**Method**	**Number of fingers**					
		1	**2**	**4**	**6**	**8**	**10**
Case 1 - $W_B = 1$,$W_F = 1$, $L_B = 10$,$L_F = 10$	**Proposed EM**	10	23.75	71.25	145.42	246.25	373.75
	COMSOL	10	23.78	71.33	146.50	245.10	375.88
	Error	0.00%	0.08%	0.11%	0.74%	0.47%	0.57%
Case 2 - $W_B = 1$,$W_F = 1$, $L_B = 20$,$L_F = 10$	**Proposed EM**	15	41.67	135	281.67	481.67	735
	COMSOL	15	41.59	136.28	279.80	486.41	738.12
	Error	0.00%	0.18%	0.94%	0.67%	0.98%	0.42%
Case 3 - $W_B = 1$,$W_F = 1$, $L_B = 10$,$L_F = 20$	**Proposed EM**	15	29.17	77.5	152.5	254.17	382.5
	COMSOL	15	29.42	77.19	152.07	257.51	385.73
	Error	0.00%	0.85%	0.41%	0.28%	1.30%	0.84%

are provided in branches a and b. Again, we observe that the obtained results are also very close to COMSOL results and the average error rate is 0.56% while the maximum error is 1.42% in case 2 reverse. Also, it can be seen if the total length of branches a and b increases, the stress increases. Furthermore, the location and current density of the stub can have a significant impact on the stress at the stub (branch c). With zero current density, the stub can decrease the stress if it is closer to the cathode, or if its length is decreased. Also, if the stub is placed further away from the cathode and its length is longer, the stress increases.

3.2.3 Results for comb structure interconnects

Now, we further validate the proposed VBEM method on the comb-structured interconnect. We list the parameters used for different test cases and the predicted stress and error rate in Table. 3.8. Fig. 3.19 shows the impact of the number of fingers, N, on the stress at the node 0 for these three cases or configurations. As we can see, with an increase in N, the EM stress increases super-linearly. Besides, we can observe from Fig. 3.10 that, increasing L_B and L_F increases EM-induced stress at node 0. However, the increase in L_F only has a small effect on EM stress as compared to the increase in L_B. Furthermore,

the results of the VBEM approach show good agreement with the results obtained from COMSOL. The average error is 0.49% and maximum error is 1.3% in case 3 at N=8.

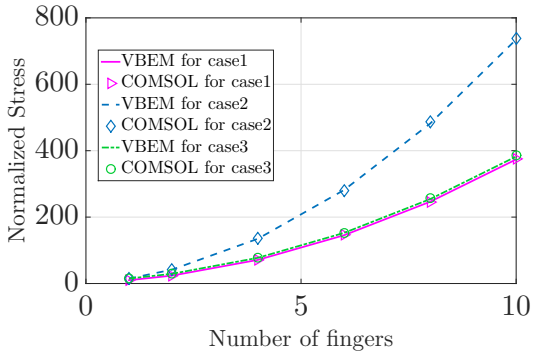


Figure 3.19: EM stress validations for each comb structure interconnect case (x: number of fingers, y: EM stress at the node 0 (cathode node))

3.3 Summary

In this chapter, we have presented a novel and fast EM immortality check for general multi-branch interconnect trees. The new method estimates the EM-induced steady-state stress in general multi-segment copper interconnect wires based on a novel parameter, *Critical EM Voltage,* $V_{Crit,EM}$. We have shown that the $V_{Crit,EM}$ essentially is the natural, but important extension of the *Blech product* or *Blech limit* concept, which describes the EM immortality condition for a single-segment wire, to more general multi-segment interconnect wires. Furthermore, the new VBEM analysis method is very amenable for EM violation fixing as it brings new design knobs and capabilities into the physical design flow. The resulting EM risk assessment method can be much easier to integrate with physical design tools and flows. The new voltage-based EM stress estimation method is based on

the exact solution of fundamental steady-state stress equations. We have shown that the proposed VBEM analysis method agrees with the results from the finite difference method in the steady-state through one example and also agrees with one published closed-form expression of steady-state stress for a special 3-terminal wire case, which further validates the proposed method. Furthermore, we compared VBEM against the COMSOL finite element analysis tool and another published EM numerical simulator XSim and it was shown that the VBEM approach agrees with both of them very well in terms of accuracy. We also studied the impact of the current crowding of practical interconnect wires on the estimated steady-state stress and showed that the effect is not significant if the length of the wire length is much greater than its width. An extension of the VBEM method to consider the significant current crowding effects was given and additionally, we analyzed mesh-structured interconnect wires and demonstrated that the proposed VBEM method is correct and accurate on these structures as well.

Chapter 4

Saturation Volume Estimation For Multi-Segment Copper Interconnect Wires

In this chapter, we present a new formula for fast estimation of the void's saturation volume for general multi-segment interconnect wires. The new formula is based on the fundamental atom conservation at the steady-state condition of void growth phases. The new void saturation estimation formula agrees with the existing single segment wire saturation void volume formula and is an natural extension of the single segment case to general multi-segment wires. In addition, we consider impacts of the void volume on final stress distributions of the wire to further improve the accuracy of the proposed formula. The proposed saturation void volume estimation can be applied for fast EM immortality check for nucleated wires, which were considered to be failed wires in the past. Combined

EM immortality check for nucleation [20,21] mentioned in chapter 3, we propose a new EM immortality check algorithm, which considers both void nucleation and void growth for the first time and thus overcoming the conservativeness of the existing EM immortality check method. Our numerical results show that the proposed formula agrees well with a published work for two-segment cases, which are supported by experimental data. The formula is also validated by the recently proposed physics-based 3D finite element (FEM) analysis tool for general multi-segment interconnect wires. We also demonstrate new EM immortality check flow can quickly identify the new type of immortal wires, which are nucleated but with smaller-than-critical voids.

4.1 The new void saturation volume estimation for general multi-segment wire

4.1.1 Review of the void saturation volume for single segment

As previously mentioned, computing the saturation void volume at steady state is critical for the immortality check of a wire. After a void is formed in a segment, the tensile (positive) stress around the void will gradually reduce to zero and the stress distribution will become compressive (negative) as shown in Fig. 4.1.

From the physics perspective, once a void is formed, the void volume $V_v(t)$ in a multi-segment wire will satisfy the following atom conservation equation [25].

$$V_v(t) = -\int_{\Omega_L} \frac{\sigma(t)}{B} dV \tag{4.1}$$

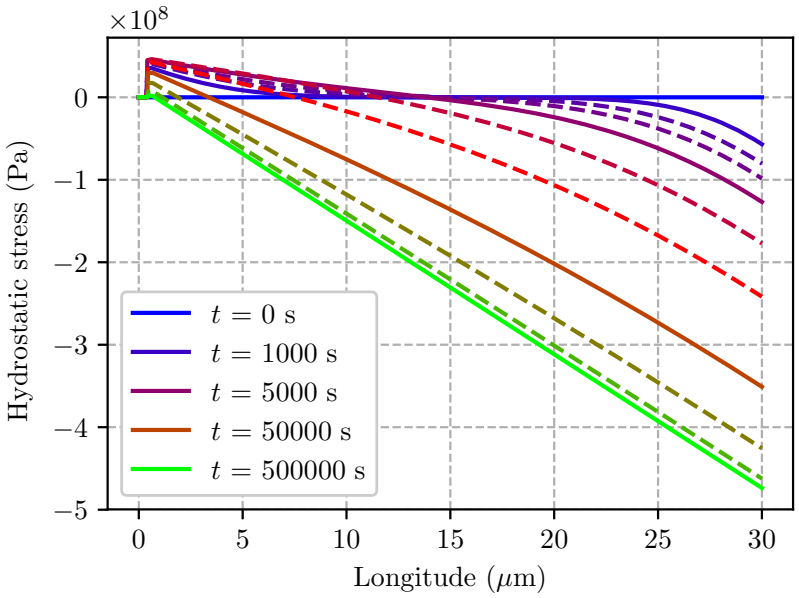


Figure 4.1: The typical stress evolution on a 30 μm copper wire computed by finite element analysis.

where Ω_L is the volume of the remaining interconnect wire. Here W is width of wire and h is thickness of the wire. In the steady-state, the stress is compressive and negative. As a result, there is a negative sign in (4.1). For the one dimensional, single segment case with wire length L, the steady state saturation volume (V_{sat}) of the void becomes

$$V_{sat} = -A \int_0^L \frac{\sigma(x)}{B} dx = -\frac{A\sigma_{max} L}{2B} \tag{4.2}$$

where σ_{max} is the maximum stress in steady state and A is its cross section area which can be calculated by $A = W \times h$. Since length of the void is typically much smaller than the length of the wire (smaller than 1% of the segment), total length L is used here instead of the length of the remaining interconnect without the void. Another formula considering the length of remaining interconnect will be discussed in the next subsection.

If we only consider the one segment wire in the one-dimensional case as shown in Fig. 4.2(a), Eq. (4.2) essentially says that the saturation volume depends on the product of σ_{max} and L. If we plot the stress versus the length for this one segment case, the product actually is the area $A1$ in Fig. 4.2(b). Such observation is important as it will lead us to the solution for the multi-segment case shown later.

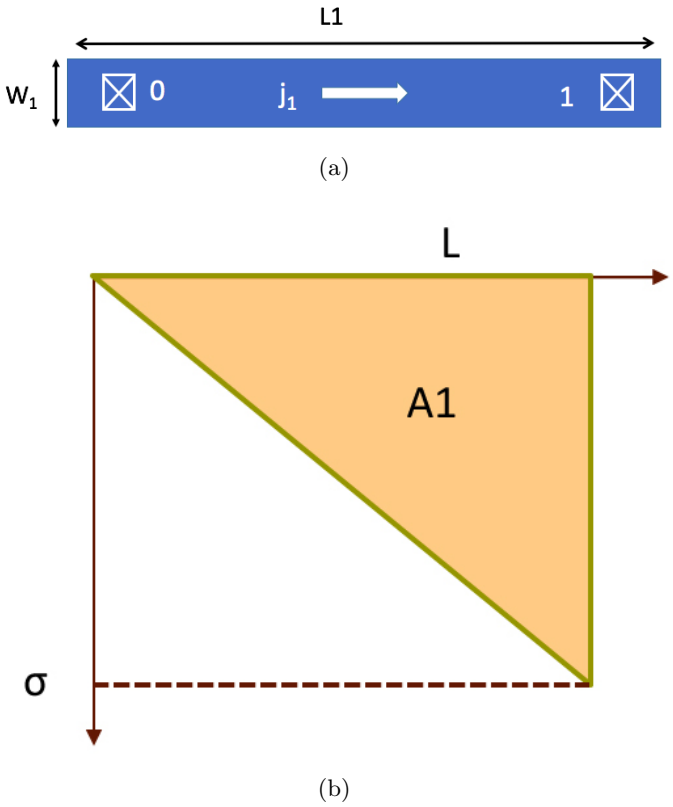


Figure 4.2: (a) A one-segment wire and the direction indicate electron flow (b) Stress integration area of a one-segment wire

To compute σ_{max}, we need to go back to Korhonens equation, in the steady state, in which we have

$$\begin{aligned} \frac{\partial \sigma}{\partial x} =& \frac{\sigma_{max}}{L} = -\frac{j\rho e Z^*}{2\Omega} \\ \sigma_{max} =& -\frac{j\rho e Z^* L}{2\Omega} = -\frac{\mathcal{V} e Z^*}{2\Omega} \end{aligned} \tag{4.3}$$

where $\mathcal{V} = j \times L \times \rho$ is the branch/node voltage between cathode and anode of the wire which was also used to represent the immortality condition here. For a single wire segment, the Blech's criterion or Blech limit, is defined as $(jL)_{crit} = \frac{\Omega(\sigma_{crit} - \sigma_{init})}{eZ\rho}$, where σ_{crit} is the critical stress and σ_{init} is the initial stress in the wire. As we can see that the immortality condition of a wire ($jL < (jL)_{crit}$) depends on the product of j (current density) and L (length). The Blech limit actually can also be written as $(jL)_{crit} * \rho = \mathcal{V}_{crit} = \frac{\Omega(\sigma_{crit} - \sigma_{init})}{eZ}$. As a result, the immortality criteria can be further written in terms of critical branch voltage $\mathcal{V}_{crit}$. Recent study shows that using the branch voltage for immortality check will become much more convenient than current density for the multi-segment case as shown in [20, 21].

Using (4.2) and (4.3), we get

$$V_{sat} = \frac{Aj\rho e Z^* L^2}{2\Omega B} = \frac{A\mathcal{V} e Z^* L}{2\Omega B} \tag{4.4}$$

where A is the cross-section area of the wire: $A = W \times h$. If we have the initial stress distribution, then

$$V_{init} = \frac{A\sigma_{init} L}{B} \tag{4.5}$$

Therefore, void saturation volume, V_{sat}, for a single wire can be expressed as

$$V_{sat} = A(\frac{\sigma_{init}L}{B} + \frac{j\rho eZ^*L^2}{2\Omega B}) = A(\frac{\sigma_{init}L}{B} + \frac{\mathcal{V}eZ^*L}{2\Omega B}) \tag{4.6}$$

which agrees exactly with [38]. However, this method only works for one dimensional single wires.

4.1.2 Proposed void saturation volume for multi-segment wires

In this work, we propose a formula to estimate the saturation volume for general multi-segment interconnect wires where each wire segment may have different widths.

Before we present our work, we would like to remark that void volume depends on many factors such as shape, location. The void location and shapes are stochastic in nature and depend on the capping and barrier materials used and the fabrication process. The void shape also keeps changing over time and the void itself may migrate as well during the growth process [39, 40]. As a result, It is very difficult to completely consider all those effects. As with all the modeling work, we have to make some assumptions based on the fundamental physics of EM effects and wire structures. Specifically, in our work, we assume that the void has a square shape initially and eventually will grow to occupy across-section of the wire with moving edges toward the atom migration direction. Eventually the void shape will become the shape of a wire or via section [12]. For the void position, statistically, the void most likely nucleate in or close to the cathode node of the wire. For the copper dual damascene wire structure, if the via is not blocked by the void, the resistance does not

increase. So the volume above the via can be recognized as the critical void volume (void fatal volume) [41, 42].

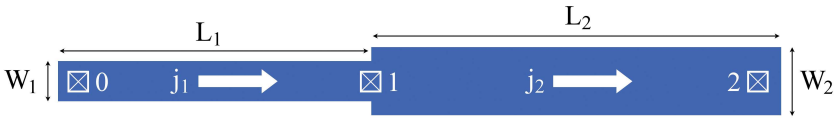
As mentioned above, void is typically formed around the cathode node of the wire. Since tensile stress at cathode is the highest part of the wire, stress at this position exceeds the critical stress first and void formation starts here. However, experimental data indeed shows that voids can also form somewhere near the cathode nodes [43]. But the location of the voids will not significantly affect the saturation volume estimation as the steady-state compressive stress distributions of the wires will be similar if the void location with zero stress is not far away from the cathode node. If void volume occupies the significant portion of the wire, zero stress location may not be very close to the cathode. But in this case, the wire will be mortal anyway and the estimation of saturation volume is less important.

Based on the previous observation, for the multi-segment case, based on (4.2), the saturation volume will be equal to the area under the multi-segment wires ($A1 + A2$) as shown in Fig. 4.3 for a two-segment case with equal widths. As a result, the saturation volume can be boiled down to computing those areas as shown below:

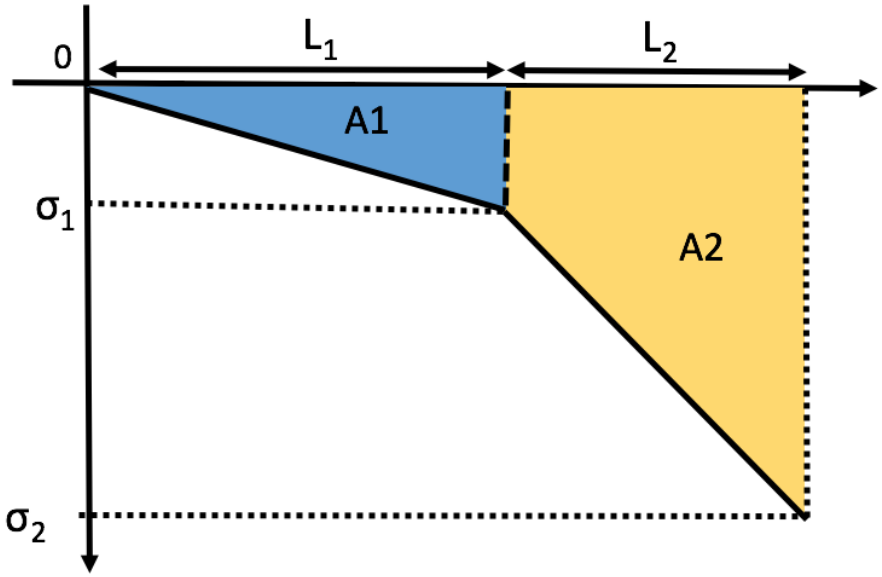
In general, for a single segment (such as $L_1$from Fig. 4.3(a)), the stress between cathode and anode can be expressed as

$$\sigma_c - \sigma_a = \frac{(\mathcal{V}_a - \mathcal{V}_c)eZ}{\Omega} = \frac{jL\rho eZ}{\Omega} \tag{4.7}$$

where σ_c and σ_a represent stress on cathode and anode respectively, and $\mathcal{V}_c$ and $\mathcal{V}_a$ represent voltages on cathode and anode respectively. At the steady state , the stress is linearly distributed on the metal wire as shown by the shaded areas in Fig. 4.3(b). Since we need



(a)



(b)

Figure 4.3: (a) A two-segment wire and the direction indicate electron flow (b) Stress integration area of a two-segment wire

to consider the width of each segment, which may be different, the problem becomes a 2 dimensional stress-area integration problem. Then the contribution from the steady-state void volume, $V_{sat,i}$, for a segment i, can be found by computing the stress-areas (width

times length of the wire):

$$\begin{aligned} V_{sat,i} &= h \times ((-\sigma_{c,i}) + (-\sigma_{a,i})) \times \frac{L_i W_i}{2B} \\ &= h \times (-2\sigma_{c,i} + \frac{\mathcal{V}_i eZ}{\Omega}) \times \frac{L_i W_i}{2B} \\ &= h \times (-2\sigma_{c,i} + \frac{j_i L_i \rho eZ}{\Omega}) \times \frac{L_i W_i}{2B} \end{aligned} \tag{4.8}$$

where $\mathcal{V}_i$,j_i,L_i,W_i are the voltage difference between anode and cathode, current density, length, and width of ith segment respectively. h is the thickness of the wire, which is the same for all the wire segments. $\sigma_{c,i}$ is the steady state stress on the cathode of segment i, which becomes 0 where the void is nucleated. We note that except for the segment with the void, steady-state stress on the cathode node of other segments are actually the anode of the segment connected to them.

Now we are ready to calculate the total void volume at the steady-state, which essentially is the void saturation volume. Given the steady-state void volume contribution from segment i, $V_{sat,i}$ in (4.8) . The total void volume can be calculated by adding all the contributions together based on (4.1). Then we have the following results:

Proposition 1. *For a general multi-segment wire, assume that a void is formed in the cathode node of one of the segments and all the initial stress' are zero. Then the saturation volume of the void $V_{sat,total}$ can be computed as*

$$\begin{aligned} V_{sat,total} = \sum_i V_{sat,i} &= h \times \sum_i (-2\sigma_{c,i} + \frac{\mathcal{V}_i eZ}{\Omega}) \times \frac{L_i W_i}{2B} \\ &= h \times \sum_i (-2\sigma_{c,i} + \frac{j_i L_i \rho eZ}{\Omega}) \times \frac{L_i W_i}{2B} \end{aligned} \tag{4.9}$$

where $V_{sat,i}$ defined in (4.8) *represents the saturation volume contribution from the ith segment. If a non-zero initial stress is considered, we can add the initial stress contributions as shown in* (4.6). □

Eq. (4.9) shows that the saturation volume is a quadratic function of wire segment lengths and a linear function of wire width in the interconnect trees. As a result, the saturation volume can be adjusted or controlled by wire width and length sizing. $V_{sat,total}$ is smaller than the critical volume, resistance will not increase. So the interconnect tree can be considered as immortal in this case. And the interconnect tree would fail if the saturation volume is larger than the critical volume.

However, since a void is formed at the cathode of the main branch, the length of the main branch is reduced. In order to have better accuracy, a reduced main branch with length of $\hat{L}_m$ is used in the calculation. Since the final saturation volume affects the remaining length of the main branch, which further impacts the final saturation volume, iterations are required to compute the final results. But for our method, we only iterate once, which is sufficient as we show later. Thus a new formula is as follows:

$$\begin{aligned}
\hat{L}_m &= L_m - \frac{V_{sat,total}}{hW_m} \\
\hat{V}_{sat,total} &= h \times \sum_{i \neq m} (-2\hat{\sigma}_{c,i} + \frac{\mathcal{V}_i eZ}{\Omega}) \times \frac{L_i W_i}{2B} \\
&\quad + h \times (\frac{\mathcal{V}_m eZ}{\Omega}) \times \frac{\hat{L}_m W_m}{2B} \\
&= h \times \sum_{i \neq m} (-2\hat{\sigma}_{c,i} + \frac{j_i L_i \rho eZ}{\Omega}) \times \frac{L_i W_i}{2B} \\
&\quad + h \times (\frac{j_m \hat{L}_m \rho eZ}{\Omega}) \times \frac{\hat{L}_m W_m}{2B}
\end{aligned} \tag{4.10}$$

Note that the $\sigma_{c,i}$ also changes to $\hat{\sigma}_{c,i}$ since the stress on main branch changes. In the following, we go through a few example to illustrate the new formula (4.9). The first example is a three terminal wire shown in Fig. 4.3. Here, stress at node 1 and node 2 can be expressed as

$$\begin{aligned}\sigma_1 &= 0 - \frac{(\mathcal{V}_1 - 0)eZ}{\Omega} = -\frac{j_1 L_1 \rho e Z}{\Omega} \\ \sigma_2 &= -\sigma_1 - \frac{(\mathcal{V}_2 - \mathcal{V}_1)eZ}{\Omega} = -\frac{(j_1 L_1 + j_2 L_2)\rho e Z}{\Omega}\end{aligned} \tag{4.11}$$

Fig. 4.4 shows calculated stress at steady state during growth phase. The stress estimation agrees with the results in Eq. (4.11). As a result, the void saturation can be computed as:

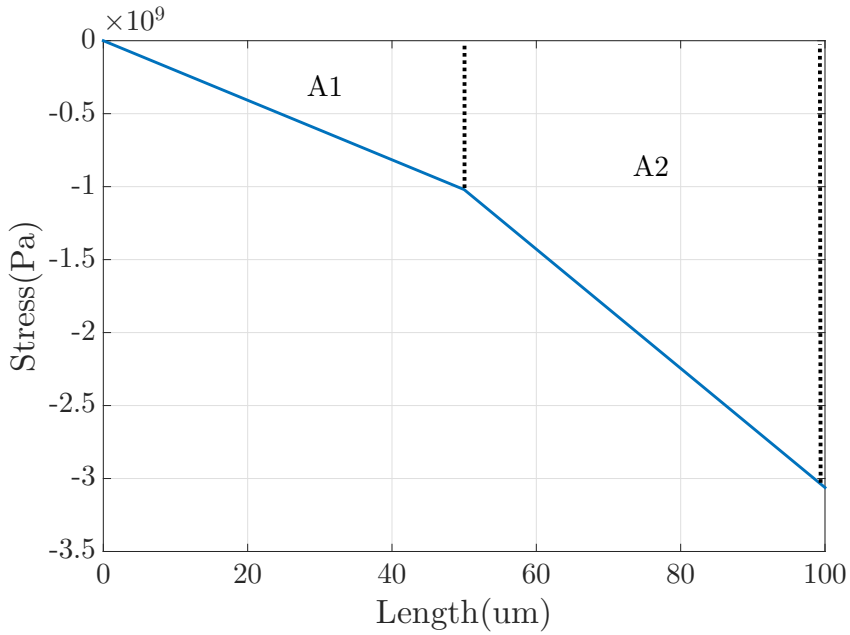


Figure 4.4: Stress distribution for two-segment wire at steady state

$$\begin{aligned}V_{sat,2seg} =& h \times \frac{-\sigma_1 L_1 W_1 + (-\sigma_1 - \sigma_2) W_2 L_2}{2B} \\ =& h \times \left(\frac{j_1 L_1^2 W_1 \rho e Z}{2B\Omega} + \frac{(2j_1 L_1 + j_2 L_2) L_2 W_2 \rho e Z}{2B\Omega}\right)\end{aligned} \tag{4.12}$$

If we consider the void volume effect using (4.10), the reduced main branch length $\hat{L_1}$ can be calculated by $V_{sat,2seg}$ in equation (4.12)

$$\begin{aligned}\hat{L}_1 &= L_1 - \frac{V_{sat}}{hW_1} \\ \hat{V}_{sat,2seg} &= h \times \frac{-\hat{\sigma}_1 L_1 W_1 + (-\hat{\sigma}_1 - \hat{\sigma}_2) W_2 L_2}{2B} \\ &= h \times \left(\frac{j_1 \hat{L}_1^2 W_1 \rho e Z}{2B\Omega} + \frac{(2j_1 \hat{L}_1 + j_2 L_2) L_2 W_2 \rho e Z}{2B\Omega}\right)\end{aligned} \tag{4.13}$$

For the other example we will consider the T-intersection shown in Fig. 4.5 In this

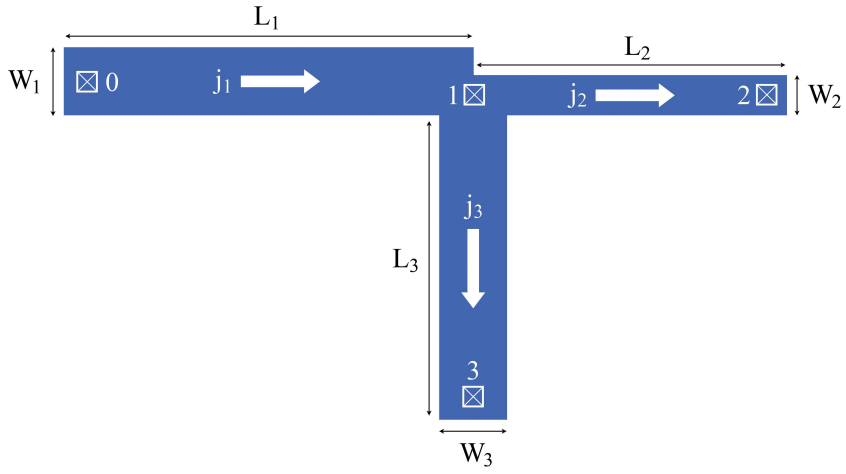


Figure 4.5: A T-shaped wire (Arrows indicate electron flow)

case a void will be formed at node 0, stress at other nodes can be calculated as:

$$\begin{aligned}\sigma_1 &= -\frac{\mathcal{V}_1 e Z}{\Omega} = -\frac{j_1 L_1 \rho e Z}{\Omega} \\ \sigma_2 &= \sigma_1 - \frac{(\mathcal{V}_2 - \mathcal{V}_1) e Z}{\Omega} = -\frac{(j_1 L_1 + j_2 L_2) \rho e Z}{\Omega} \\ \sigma_3 &= \sigma_1 - \frac{(\mathcal{V}_3 - \mathcal{V}_1) e Z}{\Omega} = -\frac{(j_1 L_1 + j_3 L_3) \rho e Z}{\Omega}\end{aligned} \tag{4.14}$$

Fig. 4.6 shows stress at steady-state during the growth phase.

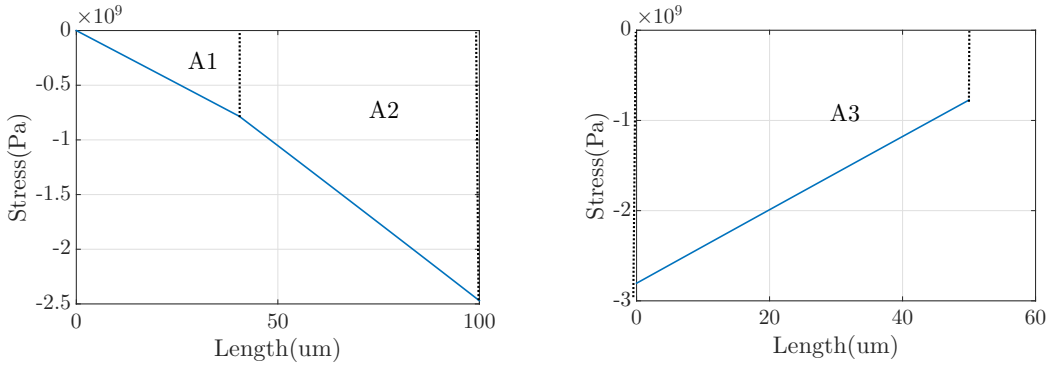


Figure 4.6: (a) Stress on horizontal segment 0-2; (b) Stress on vertical segment 1-3

Here, the saturation void volume can be calculated as

$$\begin{aligned} V_{sat,3seg} =& h \times \frac{-\sigma_1 L_1 W_1 + (-\sigma_1 - \sigma_2) L_2 W_2}{2B\Omega} \\ &+ \frac{(-\sigma_1 - \sigma_3) L_3 W_3}{2B} \\ =& h \times (\frac{j_1 L_1^2 W_1 \rho e Z}{2B\Omega} + \frac{(2j_1 L_1 + j_2 L_2) L_2 W_2 \rho e Z}{2B\Omega} \\ &+ \frac{(2j_1 L_1 + j_3 L_3) L_3 W_3 \rho e Z}{2B\Omega}) \end{aligned} \tag{4.15}$$

We can also consider volume effects by converting L_1 to $\hat{L}_1$ as shown in equation (4.10) to improve the simulation accuracy.

In order to validate this model, we validate the estimated saturation void volume with a recently proposed 3D FEM analysis tool [12, 44]. This tool employs stress model in [6], current density model, as well as Joule heating induced temperature model in the FEM based analysis method for void growth simulation. This model has high accuracy with the FEM analysis hence we use it to validate the model proposed in this study. The validation results are shown in section 4.3.

4.2 New EM immortality check algorithm

In this section, we propose a new EM immortality check algorithm for general multi-segment wires based on the new formula for estimating the saturation void volume discussed in the previous section.

First, we review the recently proposed EM immortality check for nucleation, called the voltage-based EM or *VBEM* method [20, 21]. In the new EM immortality check model, which can be viewed as a natural extension of the well-known *Blech product* [5] for immortality check of a single wire. The immortality of an interconnect tree can be summarized as

$$\mathcal{V}_{crit,EM} > \mathcal{V}_E - \mathcal{V}_{cat} \tag{4.16}$$

where $\mathcal{V}_{cat}$ is the voltage at cathode. $\mathcal{V}_{crit,EM}$ is the critical EM voltage defined by $\mathcal{V}_{crit,EM} = \frac{1}{\beta}(\sigma_{crit} - \sigma_{init})$ where σ_{init} is the initial stress. $\mathcal{V}_E$ is called *EM voltage*, which is computed as $\mathcal{V}_E = \frac{1}{2A}\sum_{k \neq g} a_k \mathcal{V}_k$, where $\mathcal{V}_k$ is the normal nodal voltage (with respect to cathode node g) at node k of the wire. a_k is the total area of branches connected to node k. Since generally the cathode node has the lowest voltage within an interconnect wire, we may just check the cathode node using (4.16). However, this EM immortality check does not consider the case where a void is nucleated in a wire, but the saturation void volume is less than the critical volume.

As a result, we propose following new EM immortality check algorithm: Given a new multi-segment wire W with its branch current and node voltage given, the new algorithm first checks (4.16) for the cathode node. If it passes, then the wire can be considered as immortal. Otherwise, we compute the saturation void volume V_{sat} from

(4.9) or (4.10). If $V_{sat} < V_{crit}$, the wire is still immortal. Otherwise, it is mortal. Then numerical methods are needed to compute the time to failure [9]. The whole algorithm flow is illustrated in the Fig, 4.7.

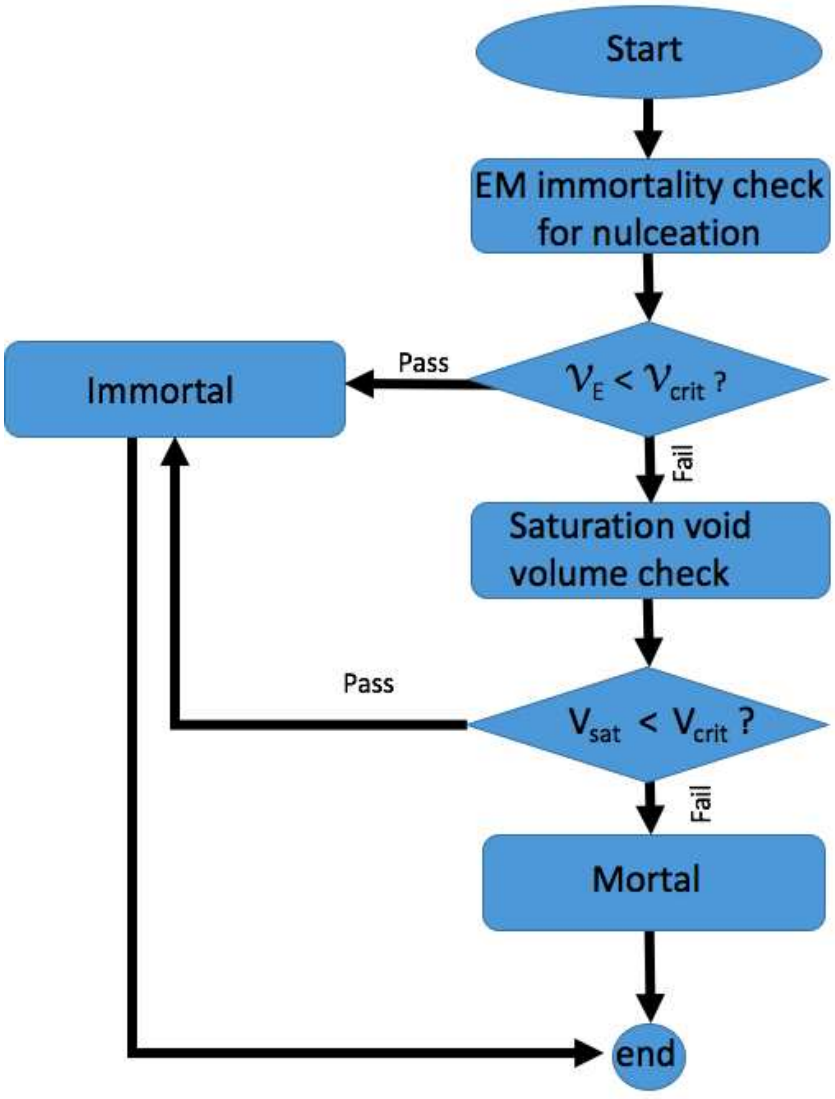


Figure 4.7: EM immortality check algorithm flow

The new EM immortality analysis flow in Fig. 4.7 can be integrated with the P/G network analysis tool for more accurate full-chip EM-aware IR drop analysis. When void volume reaches over the critical volume, the current starts to flow over the barrier layer

and wire resistance starts to increase. But for power grid networks with redundant wiring, the current indeed can start to flow alternative paths in the P/G networks. As a result, the resulting P/G networks will become a time-varying network and the wire resistance changing over time can be modeled by the proposed method, which will be very efficient (closed form expression) and more accurate than the previous method to further remove the conservativeness.

4.3 Numerical results and discussions

In this section, we validate the proposed saturation volume EM model by comparing it against other models for post-voiding process. In subsection 4.3.1, we first validate the saturation volume of the two segment wire in Fig. 4.3 using the method in [45] and the proposed method. In subsection 4.3.2 and subsection 4.3.3, we study a T-shaped wire and a more complicated 13-segment wire against the physics-based 3D FEM analysis tool [12, 44] respectively. And in subsection 4.3.4, we compare the proposed EM immortality check algorithm with the VBEM method [20, 21] using 2-segment wire test cases with different lengths and current densities.

4.3.1 Model validation on two-segment wire

For a two-segment wire structure, the saturation void volume estimation was proposed in [45] where segment 2 is treated as a reservoir ($j_2 = 0$). However, in this work, the problem is still considered as 1D where all wire segments are assumed to have the same

width. The saturation void volume, V_{max}, computed using this method is given below:

$$V_{max}/Wh = L_1 + L_2 - \frac{B}{K}[\sqrt{(\frac{Kp}{B}+1)^2 + \frac{2L_1K}{B}} - 1] \tag{4.17}$$

where $K = eZ\rho j/\Omega$. Note, V_{max}/wh is actually the saturation length. This work considers the void size formulated in the cathode of the L_1 segment and its impact on the stress distributions. In order to validate our model, we compare the void volume computed using this model against with proposed model. Our analysis shows that the void size can be small compared to the segment length and therefore negligible. Comparisons of the void volume calculated using method in [45] and proposed method are presented here. The results are shown in Table 4.1. Where V_1 is the saturation volume calculated using [45], $V_{w/o}$ is the

Table 4.1: Comparison of void area of two methods (wire width = $1um$ thickness = $0.3um$)

L_1 (um)	L_2 (um)	j (A/m^2)	V_1 (um^3) [45]	$V_{w/o}$ (um^3)	V_w (um^3)
10	10	10^{10}	0.0183	0.0184	0.0183
20	10	10^{10}	0.0485	0.0491	0.0485
10	20	10^{10}	0.0303	0.0307	0.0303
10	10	5×10^9	0.0092	0.0092	0.0092

saturation volume calculated using the proposed method without void volume effect and V_w is the saturation volume calculated using the proposed method considering void volume effect. Among the four test cases in Table. 4.1, the maximum difference is only 1.32% Between V_1 and $V_{w/o}$. And the result V_w considering void volume effect is almost the same with V_1. It can be seen that the proposed method without void volume effect is already very close to the method in [45] and if the void volume effect is considered, their results matchs exactly.

4.3.2 Model validation on a T-shaped wire

In this subsection we compare the saturation volume estimate by proposed method and a physics-based 3D FEM analysis tool [12, 44] on a T-shaped three segment wire. Four test cases with their estimated void volumes generated by the FEM tool is shown in Fig 4.8. The results of aforementioned test cases are shown in Table. 4.2.

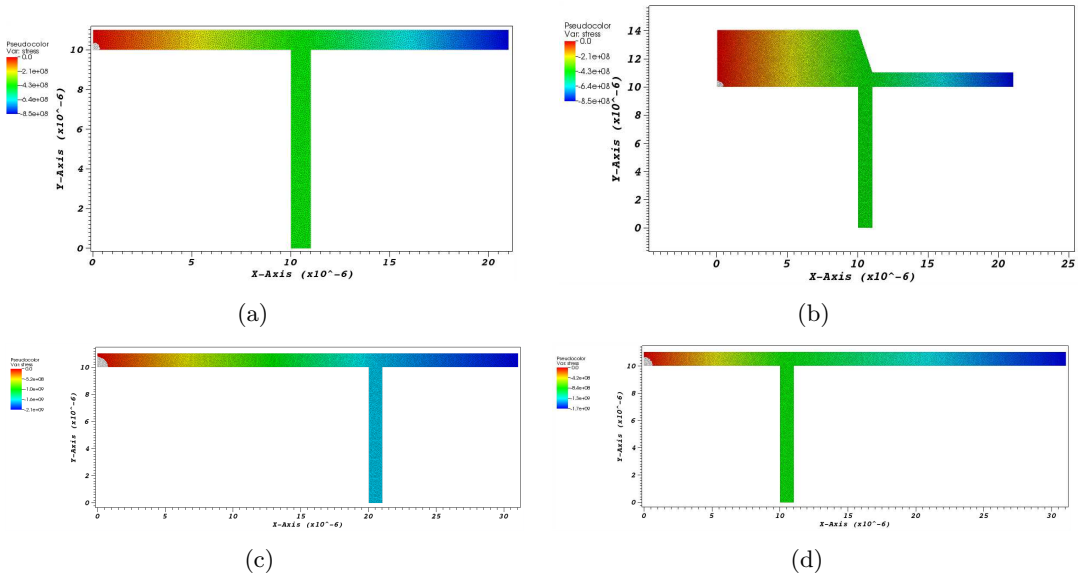


Figure 4.8: (a) Case 1; (b) Case 2; (c) Case 3; (d) Case 4

Table 4.2: Comparison of void area of two methods (wire thickness = $0.1um$)

	Case 1	Case 2	Case 3	Case 4
L_1 (um)	10	10	20	10
L_2 (um)	10	10	10	20
L_3 (um)	10	10	10	10
W_1 (um)	1	4	1	1
W_2 (um)	1	1	1	1
W_3 (um)	1	1	1	1
j_1 (A/m^2)	10^{10}	10^{10}	20^{10}	20^{10}
j_2 (A/m^2)	10^{10}	10^{10}	10^{10}	10^{10}
j_3 (A/m^2)	0	0	0	0
V_{FEM} (um^3)	0.0125	0.0185	0.0510	0.0368
V_w (um^3)	0.0121	0.0183	0.0495	0.0356

Where V_{FEM} is the saturation volume calculated using FEM analysis tool, V_w is

the saturation volume calculated using the proposed method considering the void volume effect. We can see among the four test cases in Table 4.2, the maximum difference is only 3.2% between V_{FEM} and V_w.

4.3.3 Model validation on a more complicated wire

Now we compare our estimation with the FEM simulation from physics-based 3D FEM analysis tool [12,44] on a more complicated wire structure with 13 segments as shown in Fig. 4.9. Here, the width of all the wires is 0.2 um and thickness is 0.1 um. The current

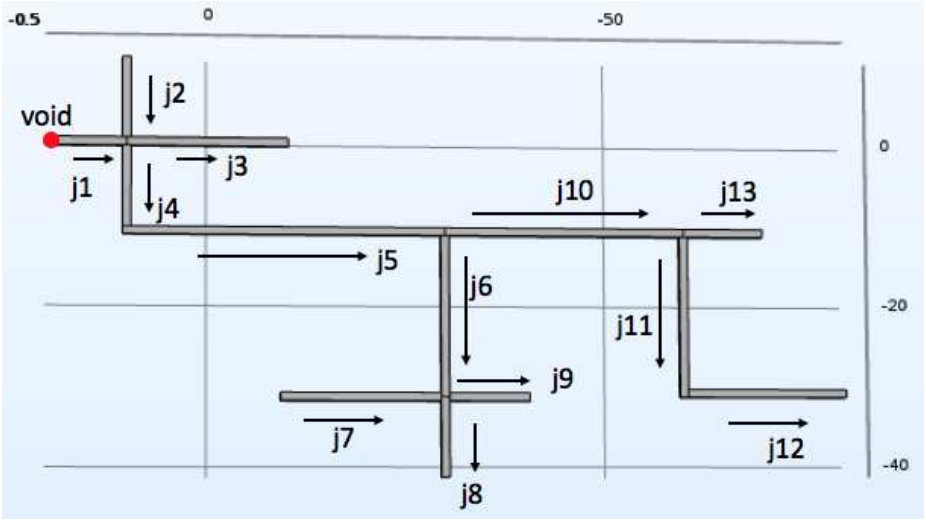


Figure 4.9: Complex multi-segment structure.

densities and lengths of all the segment are given in Table. 4.3, where $Brch$ is branch index, J is current density, Lth is the length of the wire. Critical void volume is 0.004 um^2. As shown in the Fig. 4.10, the void is formed on $segment1$.

The saturation void volume estimated by the proposed method is $0.1025um^3$ while the saturation void volume calculated by FEM analysis tool is $0.1021um^3$. As we can see, the results are almost identical (about 0.39% difference). We remark that the errors for this

Table 4.3: Parameters for the 13-segment interconnect wire

Brch#	CD (A/m^2)	Lth (um)	Brch#	J (A/m^2)	Lth (um)
1	10×10^9	10	8	15×10^9	10
2	5×10^9	10	9	5×10^9	10
3	5×10^9	20	10	10×10^9	30
4	5×10^9	10	11	5×10^9	20
5	10×10^9	40	12	5×10^9	20
6	5×10^9	20	13	5×10^9	10
7	5×10^9	20	–	–	–

complicated wire structure is much smaller than the previous T-shaped wire. The reason is that the saturation volumes for the T-shaped wire has much smaller while the 13-segment wire has much large void volume compared to the length of the wires. As a result, we obtained the much smaller relative errors for the later case.

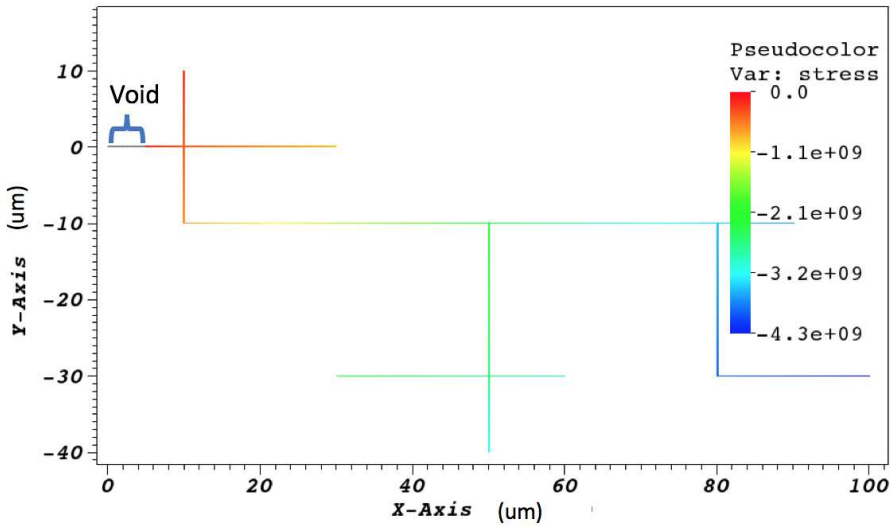
4.3.4 The results from the new EM immortality check flow

Last, not lest, we show that we can easily identify a new type immortal wires, which have voids formed and still remain immortal due to the voids being the critical void volumes. We use 2-segment wires with different lengths and current densities to perform the proposed new EM immortality check considering both saturation volume and EM immortality check for nucleation in section 4.2. The result is shown in Table. 4.4. where *VBEM* represents the

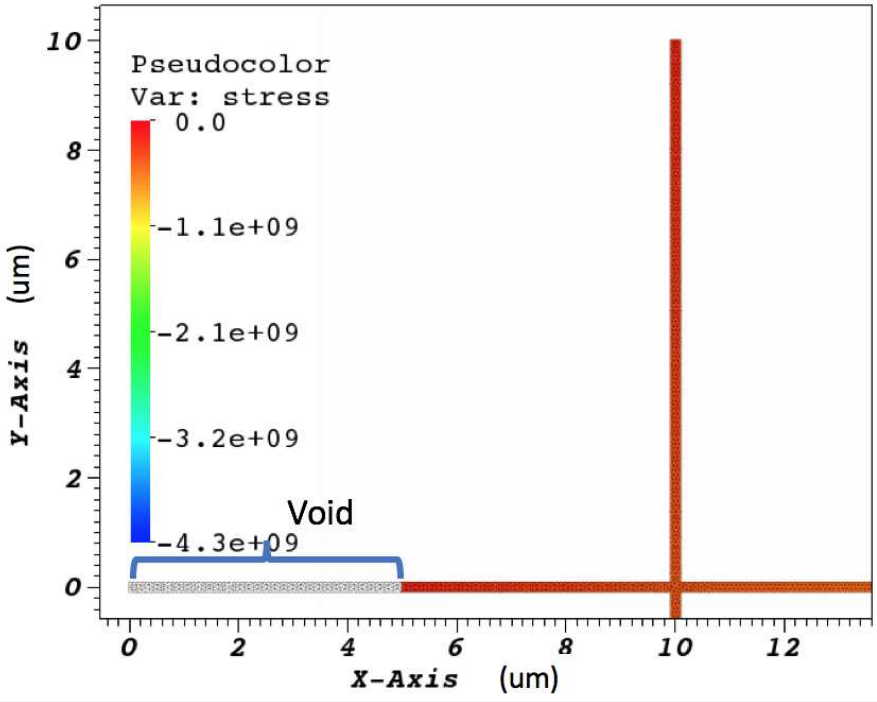
Table 4.4: Comparison of void area of two methods (wire width = $1um$ thickness = $0.3um$)

	L_1 (um)	L_2 (um)	j_1 (A/m^2)	j_2 (A/m^2)	VBEM	Proposed method
case1	10	10	10^{10}	10^{10}	Immortal	Immortal
case2	20	10	10^{10}	10^{10}	Mortal	Immortal
case3	10	50	20^{10}	20^{10}	Mortal	Mortal

EM immortality check for nucleation for general multi-segment wires [20,21]. As we can see in Table. 4.4, case 2 is recognized as mortal in the VBEM method. However, in our method,



(a)



(b)

Figure 4.10: (a) Complex multi-segment structure result from FEM analysis tool. (b) Result zoom to void area

we can see this wire is actually immortal as the void volume is smaller than the critical volume. That means the VBEM method is still too conservative and the new method can

identify more immortal cases in the EM analysis, which can translate into further reduced guard bands and reduced circuit areas and better performance.

4.4 Summary

In this chapter, we have presented a novel formula to fast estimation of void's saturation volume for general multi-segment wires. Presented new formula is less conservative than the existing method since void growth phase is also considered. The new void saturation estimation formula agrees with the existing single segment wire saturation void volume formula and is the natural extension of the single segment case to general multi-segment wires. Based on the new formula, a new EM immortality check flow is proposed including both recently proposed nucleation-based EM immortality and the void saturation volume. The new flow can further reduce the conservativeness of existing EM failure effect analysis which only considers the nucleation-based EM immortality check.

Chapter 5

EMSpice: Physics-Based Electromigration Check Using Coupled Electronic and Stress Simulation

In this chapter, we present a new full-chip EM failure analysis tool, called *EMSpice*, for physics-based coupled EM and electronic analysis. *EMSpice* takes power grid netlists from Synopsys ICC flow, and outputs the failed EM wires and their resistance changes and resulting IR drops of the power grids over the given aging time. It can be used as a EM sign-off tool or used in the EM optimization engine for commercial EDA physical synthesis flow. *EMSpice* simulator simultaneously considers the hydrostatic stress and electronic current/voltage in a power grid network. The new method starts from first

principles and can solve the resulting coupled time-varying partial differential equations in time domain to accurately stress evolution in multi-segment interconnect trees for EM failure analysis. *EMSpice* simulator employs a finite difference time domain (FDTD) solver for stress analysis for every interconnect tree for both nucleation and post-voiding phases. At the whole power grid circuit level, *EMSpice* simulator also couples the EM analysis with IR drop analysis at time domain. Thus the solver can consider the interplay among stress, void growth, resistance change and IR drop in a single simulation framework. Furthermore, *EMSpice* simulator considers both recently proposed nucleation phase immortality and incubation phase immortality for the first time to remove immortal interconnect trees from EM analysis. *EMSpice* simulator can work seamlessly with Synopsys IC (ICC) compiler physical synthesis flow. *EMSpice* simulator reads the power grid layout from Synopsys ICC and it can output the power grid layout with current density, hydrostatic stress, IR drop, and void distributions for any given specific time on the power grid layout in a graphic way for better EM failure analysis and optimization.

5.1 Related works

A few full-chip EM analysis for power grid networks have been proposed recently [11, 21, 46, 47]. Their methods can predict the EM lifetime of the power grid and obtain failed trees. Specifically, Huang *et al.* proposed first physics-EM model based full-chip EM analysis method [7, 46]. The EM model in this method only considers nucleation and growth phase. An approximate closed form obtained from analytic solution of the nucleation phase of Korhonen's equation is employed to estimate nucleation time (t_{nuc}).

Initial current densities are used for t_{nuc}. Resistance starts to increase immediately when the void is nucleated, i.e. $t > t_{nuc}$. If the time exceed the t_{nuc}, a simple linear model is used for resistance change calculation. This method indeed considers interaction between the EM and IR drops of power grids but the compact EM models are less accurate.

To get better accuracy, Chatterjee *et al.* proposed finite difference method (FDM) based full-chip EM analysis tool [11, 47]. In order to accelerate the simulation, a fitting-based immortal wire filter was applied in this work. Trees most likely to form a void were picked out and the FDM based method were only applied to solve these trees. The filter uses a conservative approximation model [48] to quickly estimate the nucleation time, t_{nuc}. If this estimated time is less than a time threshold, the trees were picked out for further EM simulation. Time threshold in this work [11] is estimated by the Monte Carlo process, specifically, is the mean of samples' TTF which has a limited normal distribution. Here the samples are from an IBM power grid. FDM was applied to solve these mortal wires and a model order reduction technique based on matrix exponential form was applied to reduce simulation costs. However, this method also mainly considers the EM stress without considering impacts from wire resistance changes of power grid networks.

Cook *et al.* proposed a FDM accelerated by Krylov subspace based reduction technique [10]. This method can be applied to general multi-segment interconnect wires with time-varying currents and temperature. To further speed up FDM, the Krylov based subspace reduction techniques in frequency domain were applied, which leads to order of magnitude speedup over plain FDM. But this method still considers the EM stress and ignores the EM and IR drop interaction in power grid networks.

5.2 The proposed new coupled EMSpice simulation for EM failure check

5.2.1 The new power grid EM check flow

Before presenting the coupled EM-IR drop analysis for full-chip EM check and analysis, we first present the proposed new EM sign-off and check flow as shown in Fig. 5.1.

The whole EM check flow mainly consists of four major steps: the *power grid generation step* from the EDA tool (Synopsys ICC), *EM immortality filtering step*, the *FDTD EM solver and linear network IR drop solver*, and the *EM check framework GUI*. In the power grid generation step, the power grid information is constructed from Synopsys IC Compiler (ICC) during the physical synthesis process for a specific design. Power grid information is dumped during power grid synthesis step in ICC flow and the P/G layout geometrical and layer, via information as well as branch currents are also dumped at the same time. After this, the power gird and corresponding branch current are first passed to the EM immortality filter step (to be discussed in subsection 5.2.3). Immortal trees are filtered out since they will never fail and resistance of these trees will never change. After this step, all the mortal trees are passed to the coupled solver step. The coupled solver consists of the FDTD for EM stress solver and linear network IR drop solver (which will be discussed in section 5.2.2). In this step, hydrostatic stress on these mortal tree wires will be simulated and void position and volume size will be calculated. All this information will then be passed to the EM check framework graphical user interface (GUI) for interactive user analysis. The EM check framework GUI can show the current density, hydrostatic

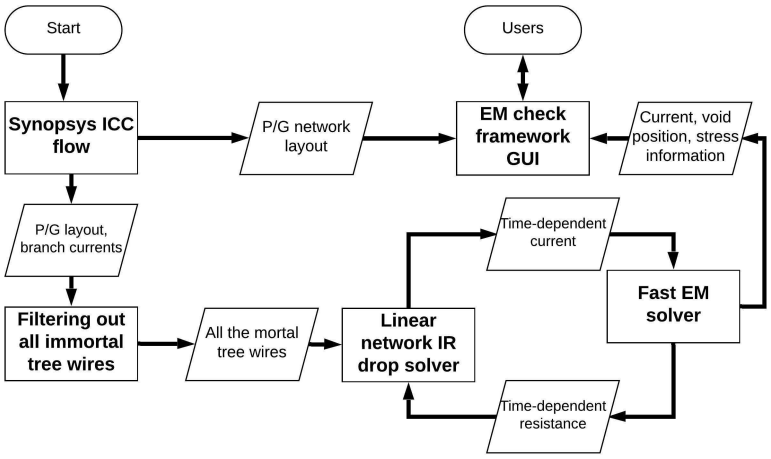


Figure 5.1: Simulation Framework for *EMSpice* simulator

stress, IR drop, and void distributions for a given specific time. In the following section, we will present computing steps in detail.

5.2.2 Physics-based coupled multi-physics power grid simulation

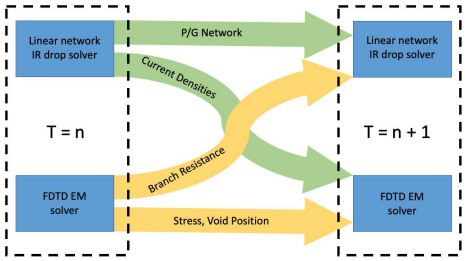


Figure 5.2: Block diagram of coupled FDTD and IR drop power grid simulator

In order to improve the accuracy of the EM stress estimation in the full-chip power grid level, we have to consider multiple physics effects and their interaction: the stress evolution modeled by (2.2), atom conservation equation for void volume, and electrical potential change over time. It was shown, when void is formed, that the void volume (shapes) and the stress distributions of the remaining wire are correlated by the following

atom conservation equation [25]

$$V_{\mathrm{S}}(t) = \frac{1}{B}\int_{\Omega_L} \sigma(V,t)dV, \quad \forall t \tag{5.1}$$

where V_{S} is void volume. As a result, we can estimate void volume or sizes once we know the stress distributions.

For electrical potential, since EM analysis is done on a time scale of at least days, we treat the electrical potential distribution u as quasi static so it can be described by the strong form Laplace equation

$$\nabla \cdot \left(\frac{1}{\rho(\phi)}\nabla u\right) = 0, \quad \text{in}\,\Omega_L, \tag{5.2}$$

$$u = g_u, \quad \text{on}\,\partial\Omega_L \cap \Gamma_u, \tag{5.3}$$

$$\vec{n} \cdot \frac{1}{\rho(\phi)}\nabla u = g_j, \quad \text{on}\,\partial\Omega_L \cap \Gamma_j \tag{5.4}$$

where $\rho(\phi)$ is the copper resistivity that is implied by the phase field, Γ_u denotes boundaries where Dirichlet (voltage) conditions are applied and g_u represent voltage/potential source, Γ_j denotes the boundaries where Neumann (normal current flux) conditions are applied and g_j represents current density source. For our full-chip EM analysis, however, it will be very expensive to solve (5.2) using the numerical approaches. Instead, we assume that homogeneous current density distribution along the wires and simple resistance network is extracted from the power grid network layout. The modified nodal analysis (MNA) is applied, which will lead to

$$\mathbf{M}(t)u(t) = \mathbf{P}I(t) \tag{5.5}$$

where $\mathbf{M}(t)$ is the admittance matrix for the power grid network, which is time-varying due to the fact that wire resistance will change with EM failure effects over time. $\mathbf{P}$ is the $b \times p$ input matrix, where p is the number of inputs or the size of driving current density sources $I(t)$. $u(t)$ represent the nodal voltages in the network and $I(t)$ are the current sources from the function blocks of the chips. From this equation, we can also compute the branch current and nodal voltages of all the branches, which are required for both immortality check and Korhonen's equation.

For Korhonen's equation, the FDTD method is carried out for each multi-segment interconnect tree for detailed stress analysis over time. We follow the similar work [10] without the model order reduction feature. After the finite difference discretization process, the partial differential equation in (2.2) with boundary conditions will be converted to the following linear time invariant (LTI) system:

$$\begin{aligned} \mathbf{C}\dot{\sigma}(t) &= \mathbf{A}\sigma(t) + \mathbf{P}I(t), \\ \sigma(0) &= [\sigma_1(0), \sigma_2(0), ..., \sigma_n(0)] \end{aligned} \tag{5.6}$$

$\mathbf{C}, \mathbf{A}$ are $n \times n$ matrices. $\mathbf{P}$ is the $b \times p$ input matrix, which is also used in (5.5). Note that $\sigma(0)$ is the initial stress at time 0. For each new simulation step, the stress from previous simulation will be used as initial condition. Now we are ready to present the final coupled

EM-IR drop partial differential equations which can be written as follows:

$$\begin{aligned} \mathbf{C}\dot{\sigma}(t) &= \mathbf{A}\sigma(t) + \mathbf{P}I(t), \\ V_v(t) &= \int_{\Omega_L} \frac{\sigma(t)}{B} dV, \\ \mathbf{M}(t) \times u(t) &= \mathbf{P}I(t), \\ \sigma(0) &= [\sigma_1(0), \sigma_2(0), ..., \sigma_n(0)] \ , at \ t = 0 \end{aligned} \tag{5.7}$$

The three equations are coupled and solved together as shown in Fig. 5.2. Linear network IR drop solver passes time-dependent current density information and P/G layout information to the FDTD EM solver. Once the voids are formed and IR drop occurs in the power grid, the current at each time step will be different. The FDTD EM solver will provide the IR drop solver with new resistance information for wires with voids. As we can see, these two simulations are coupled together, and wire current and resistance depend on each other for each mortal wires. Note that $\mathbf{C}, \mathbf{A}$ matrices which depend on wire structures are time-independent in the coupled equation.

Wire length change with void growth is also considered in the coupled simulation. The length of segment with void is updated each iteration and become shorter with void growth. This change is very small, and do not have very significant effect on the stress distribution on the tree since the void size generally is less than 1% of the total size of a tree.

Impact of resistance change of power grid networks may loosen the EM constraint significantly since current densities of the interconnects reduce. However, this effect is not thoroughly studied in aforementioned methods in section 5.1. Huang's method [7, 46]

considers this impact in the growth phase with linear model. But the accuracy of compact linear model is not high enough for resistance estimation. Besides, Chatterjee's method [11, 47] and Cook's method [10] focus on EM stress analysis and ignore the impact of resistance change caused by EM and IR drop interaction in power grid networks.

5.2.3 EM immortality filtering

FDTD based EM simulation is still computationally intensive for full chip EM analysis. With the sheer number of interconnect trees in a chip, FDTD is not efficient. However, in most designs, the tensile stress of many trees will not exceed the critical level and no void will be formed on these trees. Besides, trees that do nucleate a void do not mean they will fail since the void may not be large enough to exceed the critical void volume. If a tree is either nucleation phase immortal or incubation phase immortal, the resistance does not change. In other words, these trees will not have any impact on the IR drop and FDTD EM simulation is not necessary on these trees. We first present the immortality filter flow.

The immortality filtering flow

The immortality check flow is shown in Fig. 5.3, which consists of two filtering algorithms. First, a tree in the power grid is checked by nucleation phase filtering to see if voids can be nucleated. If no void can be nucleated, the wire is immortal. If voids will be nucleated, the tree is then passed to incubation phase immortality filter. If voids volume cannot exceed critical void volume, the tree is still treated as immortal. Only the mortal trees whose resistance will change are simulated with coupled EM-IR simulation.

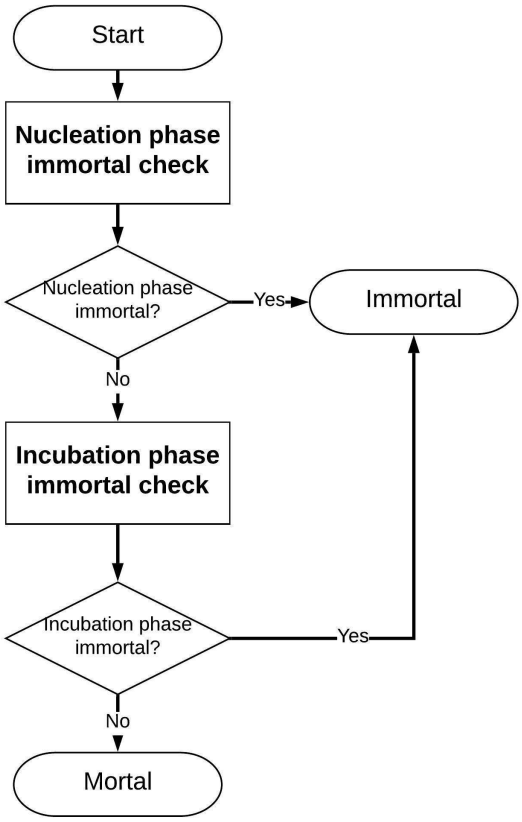


Figure 5.3: Block diagram of EM immortality filter

We remark that the Blech's product was widely used to find the immortal wire before. However, it only works for single-segment [5]. For multi-segment wires, an empirical curving-filtering based approximation method has been proposed [11, 47] which is discussed in section 5.1. However, the accuracy of this method depends on the training of the models. In contrast, the proposed EM filtering method is based on physics-based models for the immortality and thus is more accurate and predictable for over a wide range of stress conditions and technology nodes. Also we consider both nucleation and incubation immortal cases for the first time. In the section, we briefly discuss the two filtering algorithms.

Nucleation phase immortality filtering

If the stress on the cathode of a tree, at steady state (σ_{steady}) , is lower than the critical stress (σ_{crit}), then the tree is considered to be *EM nucleation phase immortal*. Based on the well-known steady state analysis method *Blech product* [5], if the current density and the wire length are not large enough, σ_{steady} cannot exceed σ_{crit}. However, this method is only suitable for a single wire with only one branch. Recently, a voltage-based EM nucleation phase immortality analysis method for multi-branch interconnect trees has been proposed [20, 21].

In this method, an *EM voltage* ($\mathcal{V}_E$) which is proportional to stress at the cathode node (σ_c) is calculated as

$$\mathcal{V}_E = \frac{1}{2A} \sum_{i \neq c} a_i \mathcal{V}_i \tag{5.8}$$

where $\mathcal{V}_i$ is the normal nodal voltage (with respect to cathode node c) at node i, a_i is the total area of branches connected to node i and A is the total area of the wire. That equation is derived from the equation for atomic conservation equation. Relationship between σ_c and area of segments as well as corresponding nodal voltages is shown in that equation. With the voltage of the ith node($\mathcal{V}_i$), steady state stress at that node (σ_i) can be calculated as $\sigma_i = \beta(\mathcal{V}_E - \mathcal{V}_i)$, where $\beta = \frac{eZ}{\Omega}$, e is elementary charge. A *critical EM voltage* $\mathcal{V}_{crit,EM}$ is defined by

$$\mathcal{V}_{crit,EM} = \frac{1}{\beta}(\sigma_{crit} - \sigma_{init}) \tag{5.9}$$

where σ_{init} is the initial stress.

EM failure will not happen if cathode node can pass the immortality check since it has the lowest voltage among all the nodes on a tree. Following equation is the condition of immortality of the tree:

$$\mathcal{V}_{crit,EM} > \mathcal{V}_E - \mathcal{V}_{cat} \tag{5.10}$$

where $\mathcal{V}_{cat}$ is the voltage at the cathode.

Incubation phase immortality filtering

Voids are formed on trees that do not pass the nucleation phase immortality. After a void is formed, it will keep growing until saturated. Void saturation happens when two kinds of flux balance with each other. One is the flux of atoms previously located in metal which is consumed by the growing void and the other is the back flux of atoms generated by a gradient of growing stress. If the void volume is smaller than the volume of the intersection, which is recognized as critical volume V_{crit}, current can still flow through the copper wire as the wire cross-section is not blocked by the void. Once the void size is large enough and occupies the wire cross-section, current has to go through the liner whose resistivity is much higher than copper and the resistance of the wire will increase. As can be seen, if the saturation volume is smaller than critical volume, the wire is still immortal although a void is formed. This case is called *EM incubation phase immortal*.

In order to determine if a wire is immortal, a model predicting the saturation volume V_{sat} was proposed in [49] shown as following

$$
\begin{aligned}
V_{sat} &= \sum_i V_{sat,i} = h \times \sum_i A_{sat,i} \\
&= h \times \sum_i [(-2\sigma_{c,i} + \frac{j_i l_i \rho e Z}{\Omega}) \times \frac{l_i w_i}{2B}] \\
&= h \times \sum_i [(-2\sigma_{c,i} + \frac{\mathcal{V}_i e Z}{\Omega}) \times \frac{l_i w_i}{2B}]
\end{aligned}
\tag{5.11}
$$

where h is thickness of the wire and $V_{sat,i}$, $A_{sat,i}$, $\sigma_{c,i}$, $\mathcal{V}_i$, j_i, l_i and w_i represent the contribution to void volume, contribution to void area, stress at the cathode, current density, voltage and length and width of the ith segment respectively. For the segment in which a void has nucleated, $\sigma_{c,i}$ is 0 on the cathode where the void is nucleated. Except for the segment with the void, steady-state stress on the cathode of other segments are the same as the anode of the segment connected to them.

As shown in the Eq. (5.11), voltage and width on each branch i can contribute to the void volume. So saturation void volume can be adjusted in order to reach incubation immortality by modifying the voltage and width of the branches.

In order to know if the wire is EM incubation phase immortal, saturation volume is compared with the critical volume, specifically, it should satisfy the following equation

$$
V_{crit} > V_{sat} \tag{5.12}
$$

We remarked that our incubation phase filtering is based on the assumption that void is nucleated in cathode node, which has largest tensile stress. However, for cases that

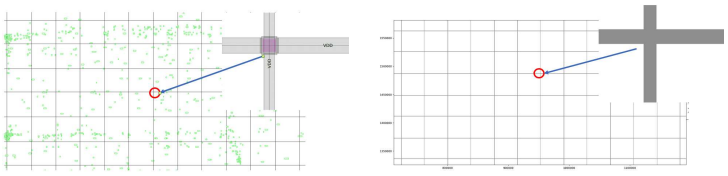
void nucleated outside cathode nodes, the immortality study in this case needs to be future investigated.

5.3 Numerical results and discussions

In this section, we present the numerical simulation and comparison results of the proposed *EMSpice* simulator. In our implementation, EM immortality filtering and Linear network IR drop solver are implemented in C++, FDTD EM analysis engine/solver was implemented in Matlab and EM check framework GUI is implemented in Python 3.6.0 with Matplotlib. The experiments were carried out on a Linux server with dual 3.3-GHz Xeon processors and 316GB memory.

In order to validate our work, we use two test cases. The first test case is the power grid of the Cortex-M0 DesignStart processor, named (*Cortex*). This is a 32-bit processor that implements the ARMv6-M architecture [50]. This processor is synthesized using Synopsys Design Compiler, and is placed and routed with Synopsys $32/28nm$ Generic Library [51]. The power grid of *Cortex* has two layers, and there are 68 trees in total.

The second case is also microprocessor design called *ChipTop*. *ChipTop* was provided by Synopsys Electronic Design University Program [52]. It is a low power design processor architecture, which contains cache blocks, I/O standard cells and digital standard cells. It is also synthesized using Synopsys Design Compiler, and is placed and routed with Synopsys $32/28nm$ Generic Library [51]. *ChipTop* has a 4 layers power grid with 912 trees.



(a) (b)

Figure 5.4: (a) Part of power grid of *Cortex* in Synopsys ICC; (b) Part of power grid of *Cortex* in proposed EM GUI

Fig. 5.4 shows the power grid layout of *Cortex* from Synopsys ICC and the same power grid in the EM check framework GUI (graphic user interface) system. In the Fig. 5.4(a), green boxes are standard cells and the mesh structure is the power grid. The zoomed area of this power grid is also shown in Fig. 5.4(a). Fig. 5.4(b) is the power grid from the EM check framework GUI. EM check framework GUI can also display the layout in an interactive way like Synopsys ICC. Zoomed power grid is also displayed in Fig. 5.4(b).

Power grid information obtained from Synopsys ICC is then fed into the proposed *EMSpice* simulator. For power grid network for *Cortex*, the *EMSpice* simulation takes about 67.14 second to finish. It takes 381.2 seconds for the simulation of power grid of *ChipTop*. In the following, we present more analysis and comparison results for these two cases.

5.3.1 Accuracy comparison against existing EM analysis methods

We first compare the proposed *EMSpice* method with existing EM analysis methods in terms of accuracy. We verify the accuracy of the FDTD EM analysis engine used

Table 5.1: Void size comparison between *EMSpice* (FDTD) and COMSOL on a wire from *Cortex*

Time after void formed	1 year	3 year	7 year
Void size from *EMSpice* (um^3)	3.63	7.38	12.04
Void size from COMSOL(um^3)	3.69	7.53	12.16
Error	1.65%	1.99%	0.98%

in the *EMSpice* for both nucleation and post-voiding phases on a few wire trees from the *Cortex* design.

We first compare the *EMSpice* with a published EM simulator XSim [36] for stress analysis of multi-segment wires in the nucleation phase. XSim is widely used to calculate the hydrostatic stress evolution in an interconnect which is assumed to be confined within diffusion barriers and it has been validated by the measured results. We take a tree from *Cortex* for stress comparison in nucleation phase. Fig. 5.5 shows the stress comparison between *EMSpice* and XSim at different time. Average error between these two method is only 0.53%, which is very small and can be ignored.

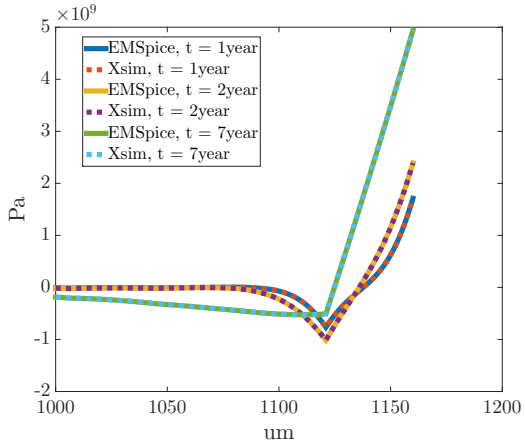


Figure 5.5: Stress comparison between the *EMSpice* and Xsim on a wire from *Cortex*
y

For EM post-voiding phase comparison, we compare *EMSpice* with the simulation result of a finite element analysis (FEA) tool, COMSOL [35] based on the same partial different equation and post-voiding boundary conditions in [24]. The void volume is also computed by using atom conservation equation (Eq. (2.6)). A 2D wire structure was used in the COMSOL simulation. Fig. 5.6 shows comparison on void size over time between

EMSpice and COMSOL, which shows that their results match very well.

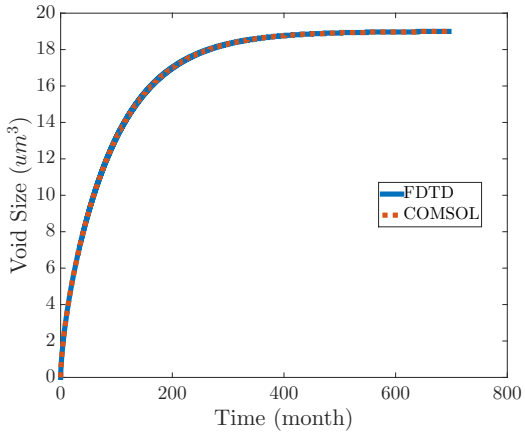


Figure 5.6: Void size comparison between *EMSpice* and COMSOL on a wire from *Cortex* design

Table 5.1 shows detailed void size comparison between the two methods on different time for the aforementioned wire in Fig. 5.6. As can be seen, the error is less then 1.99% and accurate enough for the void volume estimation.

We want to remark that Korhonen's equations for EM failure modeling is well accepted in the research communities as they are validated against the silicon data or

Table 5.2: Parameter for the EM simulation

Name	Symbol	Value
Atomic lattice Volume	Ω	$1.19 \times 10^{-29} m^3$ [53]
Critical stress	σ_{crit}	$500 Mpa$ [33]
Critical EM voltage	$\mathcal{V}_{crit,EM}$	$3.69 \times 10^{-3} V$ [21]
Critical volume	V_{crit}	depends on via size
Effective bulk elasticity modulus	B	$1 \times 10^{11} Pa$ [54, 55]
Effective charge number	Z	10 [56]
Temperature	T	373K

explain well the EM failure processes observed in experimental data in many published works [6,24,39,48,57–59]. XSim [36] was also tested against the measured results from the properly design wire test structure. As a result, comparison aginst FEM analysis based on Korhonen's equation and XSim will provide good accuracy valiation for EMSpice.

5.3.2 Filtering and coupled simulation results of *Cortex*

Firstly, efficiency of EM immortality filter is employed to reduce time of EM simulation of *Cortex*. This power grid has 68 trees in total. Among all of these trees, 42 trees are filtered out by nucleation immortality filter and 15 mortal trees among them are filtered out by incubation phase immortality filtering algorithm. So the proposed EM immortality can filter out 78% trees, which translating to accelerating the whole simulation by almost 5 times. Fig. 5.7 shows the mortal trees (highlighted with current density distribution in the whole chip layout). As we can see, most of the mortal wires are located around the peripheral of the chip.

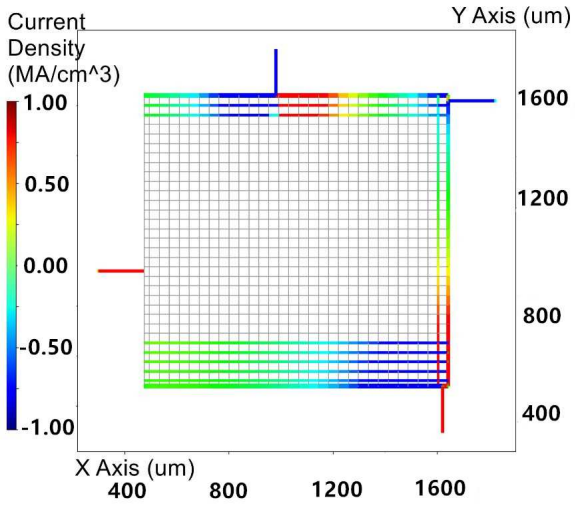


Figure 5.7: Current densities of the mortal trees. (X and Y in um and current densities in MA/cm^2)

After the filtering of the immortal wires, we start to analyze the lifetime of mortal wires in the coupled *EMSpice* simulator. As time passes, the stress starts increasing and voids are formed. Fig. 5.8 shows the stress evolution and void formation in the power grid.

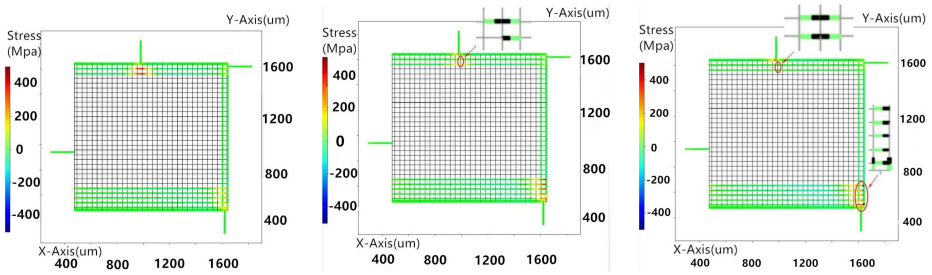
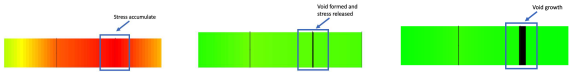


Figure 5.8: (a) Stress distribution and void formation at 8th year; (b) 12th year; (c) 20th year for *Cortex* design. (X and Y in *um* and stress in *Mpa*)

As can be seen from the Fig. 5.8 in the 8th year, all the wires are still in the nucleation phase. Stress at some hotspots has increased to a high level. In the 12th year, some voids are formed and stress near to the void starts decreasing. These wires get into incubation phase and void sizes at that time are very small and not enough to block vias cross sections. In the 20th year, more voids are nucleated and most of them are in the growth phase. Some voids have grown to the saturation volume. As aforementioned in section 2.1.2, stress accumulated on the cathode before void nucleation and decrease to zero very fast after the void formed. A set of zoomed figures of void growth process is shown in Fig. 5.9.



(a) (b) (c)

Figure 5.9: (a) Zoomed stress distribution and void formation at 8th year, (b) 12th year, (c) 20th year of *Cortex* design.

Fig. 5.10 shows the voltage drop on the full-chip power grid. As we can see, there are significant voltage drop in two areas where the void forms. It also shows that the void formation causes the resistance change and leads to large voltage drop.

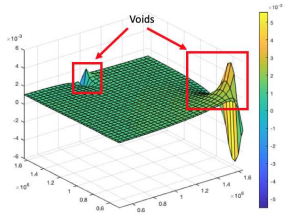


Figure 5.10: Voltage drop on the power grid of *Cortex*

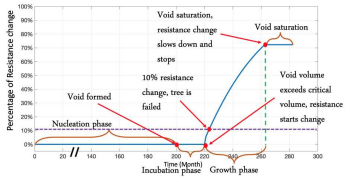


Figure 5.11: Resistance change over time on a mortal tree on power grid of *Cortex*

Fig. 5.11 shows resistance change over time of one of the mortal trees experiences the late failure effect. As shown in the figure, the void is formed at the end of nucleation

phase. However, the resistance does not change at that time. At the end of incubation phase, the void occupies the cross section of the via, and resistance starts to change since this is a late failure case. The resistance change pattern shown in Fig. 5.11 based on the three-phase EM model mentioned in chapter 2, is consistent with the experimental observation [60]. When the increment of resistance reaches 10% (note that this failure criterion is used for illustration purpose), this tree is marked as failed. Finally, the void saturates and resistance change stops. Node voltages keep changing with time after void volume exceeds a critical

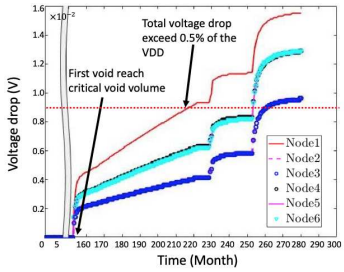


Figure 5.12: Voltage drop change over time for different nodes on power grid of *Cortex*

level. As shown in Fig. 5.12, voltage changes with void growth and a sudden voltage drop happens once a new void exceeds the critical level. Compared with Fig. 5.12 and Fig. 5.13, it can be observed that the time when sudden voltages drop matches well with the time when voids exceed critical volume.

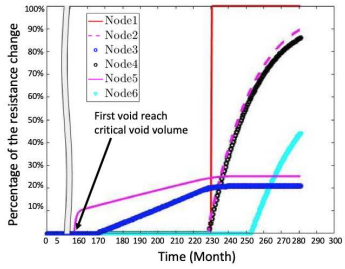


Figure 5.13: Resistance change over time of failed interconnect tree wires on power grid of *Cortex*

Fig. 5.13 shows the resistance of some failed tree wires. We can see that trees failed at different times. Both early failure and late failure can happen in this power grid. Early failure cause open circuit and resistance goes to infinite. Resistance of late failure trees increase gradually and finally stop when voids reach saturation volume.

5.3.3 Filtering and coupled simulation results of *ChipTop*

EM immortality filter is also applied for power grid of *ChipTop* which has 912 trees in total. As shown in Fig. 5.14, there are only 50 failure trees on the power grid. Among all of these trees, 695 trees are filtered out by nucleation immortality filter and 177 immortal trees are filtered out by incubation phase immortality filtering algorithm. After filtering, there are only 5.48% mortal trees which need further EM analysis. The immortality filtering dramatically reduce the simulation by about 20 times. Fig. 5.14 shows the mortal trees (highlighted with current density distribution in the whole chip layout) of *ChipTop* design. We observe that many mortal tree wires are thin wires. The reason is that these thin wires has small critical volume, although their saturation volume is not that large, they are still mortal. On the other hand, some large wires have larger critical volume. Even though their saturation volume is larger than thin wires, they are still incubation phase immortal.

Fig. 5.15 shows void formation process of *ChipTop*. Voids are not nucleated at 8th year. This time all trees are in the nucleation phase. As can be seen in the figure, stress are accumulated at cathodes. At 10th year, voids start to be formed and many wires enter incubation phase. We can see the stress near voids decreasing to zero in this figure. At 17th year, most voids are formed and some of them already reaching saturation volume.

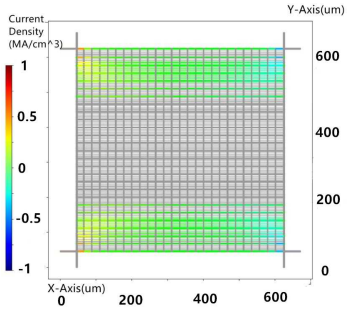
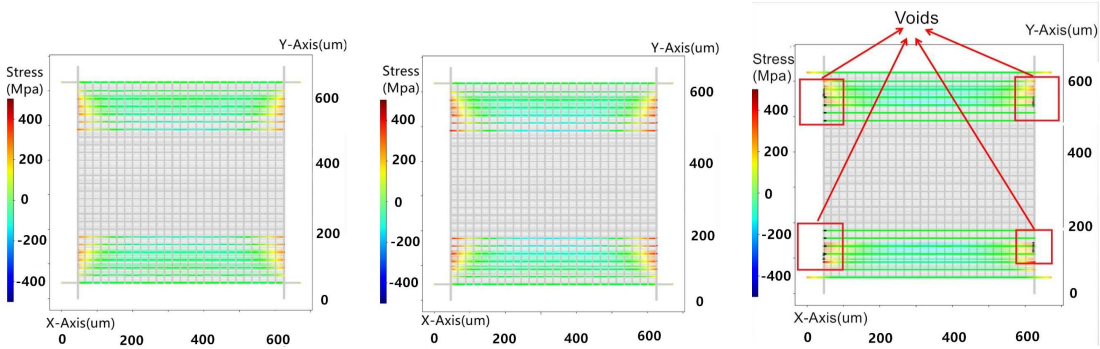


Figure 5.14: Current densities of the mortal trees on power grid of *ChipTop*. (X and Y in um and current densities in MA/cm^2)

Zoomed figures of voids are shown in Fig. 5.16. As can be seen in the figure, the voids are gathered in the edges of the power grid. Similar with the *Cortex*, the lines on the edge have larger current densities and they are more vulnerable. Based on the result of these cases, in the design, wires on edges of the power grid need to be optimized to avoid EM failure.

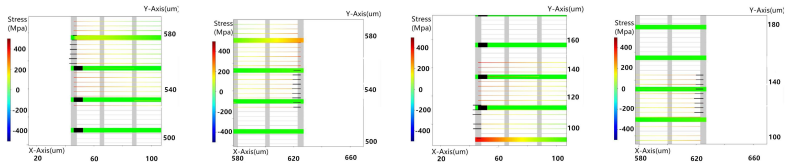


(a) (b) (c)

Figure 5.15: (a) Stress distribution and void formation on power grid of *ChipTop* at 8th year; (b) 10th year; (c) 17th year. (X and Y in um and stress in Mpa)

Table 5.3: Comparison of failed tree number of three methods on the power grid of *Cortex*

Simulation time		8 years	12 years	20 years
Black's method	Failed tree number	24	24	24
	Failed tree percentage	35.3%	35.3 %	35.3%
Huang's method	Failed tree number	11	13	16
	Failed tree percentage	16.2%	19.1 %	23.5%
EMSpice simulator	Failed tree number	0	2	9
	Failed tree percentage	0%	2.9 %	13.2%



(a) (b) (c) (d)

Figure 5.16: Void distribution on the *ChipTop* at 17th year on (a), left up corner(b) right up corner, (c) left down corner, (d) right down corner

Fig 5.17 shows the voltage change on the power grid of *ChipTop*. The voltage change significantly on the location where voids are formed. As can be seen, hill of voltages changes are in the four corners of power grid. Both early failure and late failure contribute to the voltage change in these locations. Fig 5.18 shows resistance change of some typical

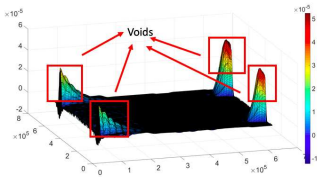


Figure 5.17: Voltage drop on the power grid of *ChipTop*

wires. In the figure, two wires are early failure and the other two wires are late failure. Here two early failure wires are with smaller current. They fail faster since the cross section of the vias on these wires are also smaller. The other two late failure wires have larger

Table 5.4: Lifetime comparison of the three full-chip EM analysis methods on the power grid of *Cortex*

Trees name	Time to Failure (years)		
	Black's method	Huang's method	*EMSpice* simulator
Tree 1	8.33	immortal	immortal
Tree 2	4.45	9.17	immortal
Tree 3	3.08	6.16	16.67
Tree 4	2.17	4.50	14.58
Tree 5	1.67	3.67	10.41

current. They are wider wires with large cross section of the vias and long incubation phases. So their resistance change happens later. As can be seen in the Fig 5.19, wires with less current have less contribution to voltage change even though they have early failure and have significant resistance change. Voltage change caused by these wires are only about 1% of the voltage change caused by wires with large current density. However, for wires with large current, there is very obvious voltage change when their resistance starts increasing.

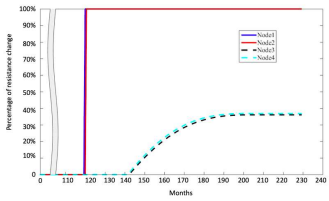


Figure 5.18: Resistance change of mortal wires on the power grid of *ChipTop*

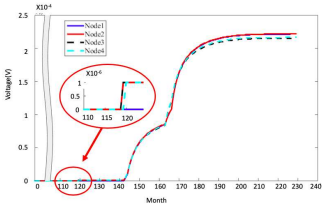


Figure 5.19: Cathode voltage drop of mortal wires on the power grid of *ChipTop*

Table 5.5: Comparison of failed tree number of three methods on the power grid of *ChipTop*

Simulation time		8 years	10 years	17 years
Black's method	Failed tree number	227	227	227
	Failed tree percentage	30.4%	30.4%	30.4%
Huang's method	Failed tree number	98	144	165
	Failed tree percentage	10.7%	15.8 %	18.9%
EMSpice simulator	Failed tree number	0	17	40
	Failed tree percentage	0%	1.9 %	4.4%

5.3.4 Comparison against existing full-chip EM analysis methods

In this section, we compare the proposed *EMSpice* simulator against one existing full-chip EM analysis method. We compare the *EMSpice* against two methods: the traditional Black's method in which the time to failure is calculated for each segment based on Black's equation [4] with the same parameters used; we also employ Huang's method [46] as another baseline.

The comparison result of power grid of *Cortex* is shown in Table 5.3 and result of power grid of *ChipTop* is shown in Table 5.5.

In the Black's method, tree failure only depends on the current densities on that tree. It cannot consider time-varying current change in the power grid. Compared with other methods, Black's method leads to most conservative results. Around 35% of the trees in the power grid of *Cortex* and 30.4% of the trees in the power grid of *ChipTop* are estimated to be failed by this method. Huang's method considers both nucleation phase and growth phase. But this method still is less accurate as compact models are used for both nucleation and growth phases. It marks 23.5% trees in the power grid of *Cortex* and 18.9% trees in the power grid of *ChipTop* as the failed trees in the 20th year. The proposed *EMSpice* simulator shows only 13.2% failed tree wires in the power grid of *Cortex* and 4.4% failed trees in the power grid of *ChipTop* in the 20th year. As we can see, at the 20th years, the number of the failed trees by *EMSpice* simulator is 63.5% less than the Black's method and 43.8% less than Huang's method in the *Cortex* case. There are 76.7% less than the Black's method and 66.7% less than Huang's method in the *ChipTop* case. As we can see, *EMSpice* method can significantly reduce the over-conservations from the existing two

methods, which can lead to more aggressive design with less guard bands, which leads to better performances under the same design costs.

Furthermore, we also look at the lifetime of some immortal and failed trees. Their lifetime given by the three methods are shown in Table 5.4. As we can see, the proposed *EMSpice* still gives the long lifetime estimation among the three methods for the five tree wires. The difference can be quite significant (up to 6.71X longer). Further, we observe that the immortal tree wire marked in the *EMSpice* method can be considered as mortal tree wires in both the Black's method and Huang's method. The reason is that the *EMSpice* method considers the unique incubation phase immortality case and it can identify the nucleated wires as immortal as long as their void sizes are small enough.

We want to remark that longer lifetime estimation for EM analysis does not alway mean better accuracy as recent work shows that Black's model can be also optimistic as well [11]. But in general, Korhonen's based EM models, more physics-based EM immortality filtering, and more accurate post-voiding analysis tend to lead to more accurate results, which typically are less conservative than existing Black-Blech based EM models and other compact models as shown in many recent works [9, 11, 46, 47, 61, 62].

Table 5.6: Lifetime comparison of the three full-chip EM analysis methods on power grid of *ChipTop*

Trees name	Time to Failure (years)		
	Black's method	Huang's method	*EMSpice* simulator
Tree 1	12.14	immortal	immortal
Tree 2	6.39	12.77	immortal
Tree 3	4.35	9.58	12.1

5.4 Summary

In this chapter, we have proposed a novel full-chip electromigration verification for power grid network of nanometer VLSI chips. The new method simultaneously considers two physics effects in interaction: the hydrostatic stress and electronic current/voltage in a power grid network. It can solve the resulting coupled time-varying (partial) differential equations in time domain to accurately predict time-to-failure for a power grid in the entire chip. The new tool reads the power grid layout from Synopsys ICC. Then an immortality filter considering both nucleation phase immortality and incubation phase immortality is applied to remove immortal interconnect trees from EM analysis. A finite difference time domain solver is employed for stress analysis for every interconnect tree. Metal atom conservation equation is used to estimate the void volume change and resistance increments over time. The EM analysis is coupled with IR drop analysis of a whole power grid networks at each time step so that we can consider the interplay among stress, void growth, resistance change, and IR drop in a single simulation framework. Accuracy of *EMSpice* has been validated by comparing with a published EM simulator, XSim, for nucleation phase, and finite element method based COMSOL for post-voiding phase. The comparison results show that *EMSpice* agrees well with both methods very well.

Chapter 6

Dynamic Reliability Management for Multi-Core Dark Silicon Processor Based on Deep Reinforcement Learning

In this chapter, we present a new deep reinforcement learning optimization frame work considering both hard and soft errors for dark silicon processors. Our work is motivated by the lack of online optimization tools and low efficiency of traditional Q-learning algorithm in run time operation. We formulate our DRM problem as minimizing the energy consumption subject to the reliability include hard and soft errors, power, performance and thermal constraints. The hard error model is based on a recently proposed physics based EM model. Hard and soft errors are obtained by recently proposed reliability models.

Power, performance are acquired by X86 based micro-architecture model [22] and thermal information is given by temperature simulators. The control operation we use dynamic voltage and frequency scaling (DVFS) and ON/OFF pulsing action. DRL method is used to select the most cost effective operation.

6.1 Review of the EM and soft error reliability models

For the system level EM analysis, we assume that each core can have its own voltage (voltage island) and frequency settings, and DVFS can be done locally for each core. For the micro-architecture of EM failure we focus on power grid since it is the most vulnerable part to EM failure. A simple mesh-structured power grid is generated for each core, and each of them has its power island with its power regulator. We assume that power grids for each core are the same and are less coupled as far as EM effects are concerned. Power for each core is estimated by a power modeling framework. For the device level power distribution, we assume that the power is evenly distributed in the power grid inside each core. Current on each branch can be obtained from the power on each interconnect tree. Lifetime of each tree is estimated by aforementioned three phase model [9] using the corresponding current and temperature. When the voltage drop caused by resistance increase of the power grid reaches a certain level, the core is considered as fail.

In order to consider lifetime impacts of different tasks, a system-level model is proposed in [63]:

$$MTTF_{EM} = \frac{1}{(\sum_{m=1}^{n}(\Delta t_m \frac{1}{MTTF_m}))/T} \tag{6.1}$$

where $MTTF_m$ is the actual MTTF (mean time to failure) under the m-th task for Δt_m period, assuming the chip works through n different power and frequency settings and $T = \sum_{m=1}^{n} \Delta t_m$.

6.1.1 Soft error reliability model considering DVFS impacts

Soft errors, or single event upset, are defined as the transient faults inside the logic or memory on a chip, and result in an incorrect system output. The soft errors can be caused by cosmic radiation, alpha particle decay, and thermal neutrons. Soft error rate (SER) is the rate in which a chip or system encounters soft errors and typically can be expressed as the number of failures in the given time. Although there is still a lack of consensus on the exact soft error rate (SER) of specific chips and systems, it is obvious that the SER per chip is practically increasing due to the increasing number of components or cores on a chip. Recently a new exponential soft error models have been introduced to account for those effects [64, 65].

For our problem, we employ an existing exponential model considering DVFS effects on soft error rate, which assumes that the radiation-induced failure follows a Poisson distribution, so the average soft error rate can be express as terms of operating frequency f, supply voltage V_{dd}, where SER_0 is the average failure rate at the maximum frequency f_{max} and voltage V_{max}, (so, $f_{min} < f < f_{max}$, $V_{min} < V < V_{max}$) in (6.2) [64].

$$SER(f, V_{dd}) = SER_0 e^{\frac{d(f_{max}-f)}{(f_{max}-f_{min})}} \tag{6.2}$$

where d is an architecture dependent constant, which is the sensitivity of failure rate with

DVFS. We also employ the previous work to model the relationship between operating frequency and supply voltage to further simplify (6.2) from [66].

$$f = \beta \frac{(V_{dd} - V_{th})^2}{V_{dd}} \tag{6.3}$$

where β is a technology-related constant, and V_{th} is the threshold voltage. By substituting (6.3) into (6.2), DVFS-aware SER equation can be derived as the function of only supply voltage V_{dd} only [65]:

$$SER(V_{dd}) = SER_0 e^{\frac{d(f_{max} - \beta V_{dd} - 2V_{th} + \frac{V_{th}^2}{V_{dd}})}{f_{max} - f_{min}}} \tag{6.4}$$

6.2 DRL framework for Dark silicon optimization

In this section, the DRL method based framework for the dark silicon will be introduced. The background of DRL method will be reviewed and the DRL method based framework applied in this work will be discussed in detail. The framework can handle high dimension state and action space and provide a fast and effective solution for energy saving of dark silicon processor.

6.2.1 Background of deep reinforcement learning

Reinforcement learning is the learning method whose goal is to map situations to actions so as to maximize the reward or minimize the penalty [67]. It can handle problems with stochastic transition without any adaptation and is a method able to converge close to the optimal solution of a state-action function (s, a) for an arbitrary policy. After each

action a Q-value ($Q(s,a)$) will be calculated with reward/penalty.

$$\begin{aligned} Q^{t+1}(s(t),a(t)) = Q^t(s(t),a(t)) + \\ \alpha(t) \times \Big(PT(t+1) + \gamma \min_a(\forall Q^t(s(t+1),a))\Big) \\ -\alpha(t) \times Q^t(s(t),a(t)) \end{aligned} \tag{6.5}$$

In Eq. 6.5, $\alpha(t)$ is learning rate between 0 and 1 for new Q-value calculation. $s(t+1)$ is the state after taking action $a(t)$, and $Q^t(s(t+1),a)$ are all possible actions' Q-values from future state. Discount factor γ (between 0 and 1) affects the importance of future penalty. $\min(\forall Q^t s(t+1),a)$ is the estimate of the optimal future value. The difference between old Q-value (Q^t) and learned value ($PT(t+1) + \gamma \min_a(\forall Q^t(s(t+1),a))$) updates the new Q-value (Q^{t+1}) with the learning rate.

As we can see in Eq. 6.5, Q-value only depends on the current state and action takes. This Q-value is stored in a large Q-table in order to determine the next optimal state. Dimension of the Q-table is ($|A||S|$) where A is a number of actions and S is a number of states since each (s,a) pair has a corresponding Q-value. And the iterations steps to get the final result would be in $O(|A||S|^2)$ [68]. As can be seen, the Q-learning method take large memory to store the Q-value and need many steps to converge to the optimized state and time to access data stored in the Q-table is long.

In order to resolve these problems, deep reinforcement learning is proposed recently in order to handle large state action space. A Q-function in Eq. (6.6) is used to estimate the Q-value of the (s,a) pair.

$$Q(s,a) \approx f(s,a) \tag{6.6}$$

As can be seen in Eq. (6.6), only when the result of the Q-function is accurate, Q-learning process can find the optimal choice. Neural network is employed so as to get the accurate Q-function and it can lead to an accurate Q-value.

In order to train the neural network, two steps are required. Data for training is required and the loss function needs to be decided. For the training parameters, learning experience of each time step is stored as (s_t, a_t, r_t, s_{t+1}). And the Q-value of each step is used as the output label of the training. That value is obtained by $r_k + \gamma max_{a'} Q(s_{t+1}, a')$ Since the training goal is to make the estimated Q-value more close to real Q-value, so the loss function can be written as [69]:

$$L = E[(Q(s,a) - Q^*(s,a))] \tag{6.7}$$

Here $Q(s,a)$ is the real Q-value and $Q^*(s,a)$ is estimated Q-value

6.2.2 Deep reinforcement learning based optimization framework

In this subsection, we formulate our new dynamic reliability management for multi-core dark silicon processor based on deep reinforcement learning. The purpose is minimizing energy considering a hard and soft error induced lifetime by controlling the p-state of cores. p-state is the performance state subject to power budget, performance deadline, and temperature constraints. The dynamic reliability management method including the DRL is

implemented in software which is used to control the hardware performance. Learning platform formulation and implementation of dark silicon evaluation platform will be discussed.

DRL method framework based formulation and solution

Based on DRL algorithm proposed in [69], a DRL method based framework is formulated in Fig 6.1.

In the first step, a learning agent is put in a random initial state. In the proposed framework for dark silicon, the state is the set of cores at different p-state. Each core has its own unique power, EM and SER. So it can be seen the power, temperature, EM and SER of the system at each states are different with different p-state of each core which leads to different penalty functions for each action which the next state agent will go to. Agent will transfer from state to state and each action will provide the agent a penalty and its goal is to minimize total penalty [70].

After initial the learning agent, transition profiles (s_t, a_t, r_t, s_{t+1}) and its corresponding Q-value $Q(s_t, a_t)$. These data are restored in an experience memory D and used for neural network training (NN). This training is done offline

With offline trained NN model, we can perform the online learning process. The agent selects the action based on a ε greedy policy. That policy is used to control the level of greed [67]. With probability $1 - \varepsilon$ select the action perform the maximum estimated Q-value $a_t = argmax_a Q(s_t, a)$. Otherwise, a random action is selected. This random selection is used to avoid the sub-optimal decisions. Whit that random selection, the learning agent will not stuck in the local optimized point and will perform global optimized result. After selected action performed, the agent takes the action and goes to the next state.

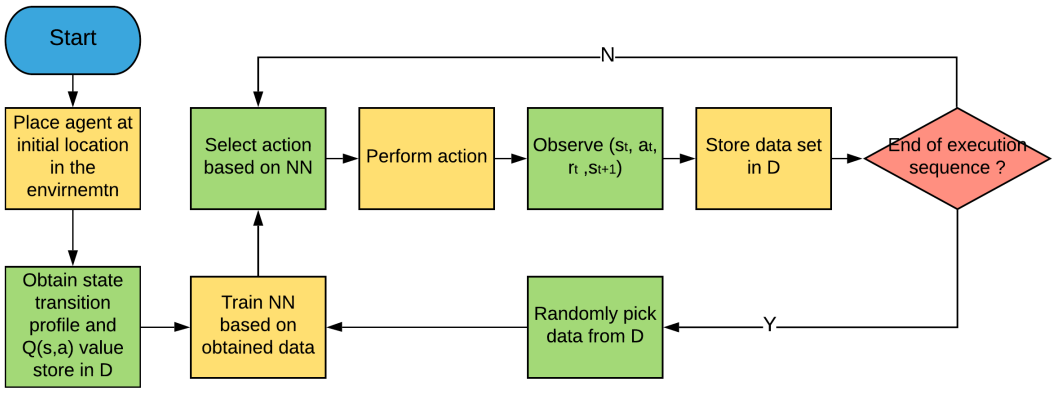


Figure 6.1: The proposed deep learning frame work for reliability management flow

After the action performed, the new transition profiles (s_t, a_t, r_t, s_{t+1}) and its Q-value $Q(s_t, a_t)$ are observed and stored in memory D. If the agent does not reach the end of an execution sequence, a new learning step is performed and a new action will be selected. Then the loop will continue. When it reaches the end of an execution sequence the NN model will be updated. Experience replay is performed and data are randomly picked from memory D. By using the experience replay the behavior distribution is averaged over many of previous state so that the learning result can be more smooth and avoid the oscillations in the output. The NN is updated with the selected data set and put back to the loop for the learning process in the new execution sequence.

Implementation of dark silicon evaluation platform

In order to evaluate the proposed DRL method based dynamic reliability, a evaluation platform is implemented as shown in Fig 6.2. In this work, SPLASH-2 benchmark [71] is used to evaluate the dark silicon multi-core processor since it includes applications and kernels in most of the area. The p-state of cores are defined by DVFS. In that state, the

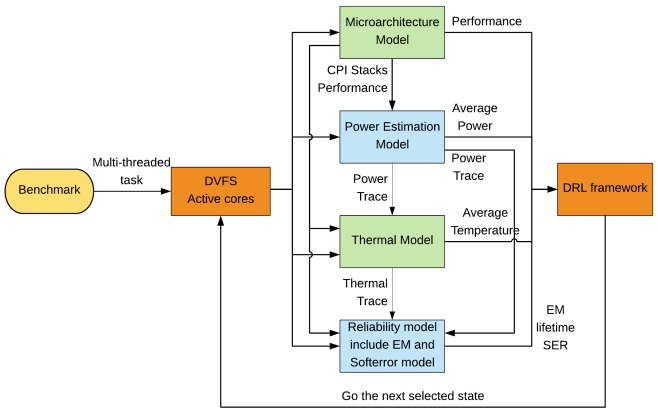


Figure 6.2: The evaluation platform for dark silicon based on DRL method

Micro-architecture simulator Sniper [72] is used to obtain the performance and Cycle per instruction (CPI) stacks. These informations are passed to power estimation model. In this work, McPAT (multi-core power, area, and timing) [73]. It can provide dynamic and static, even short-circuit power dissipation and provides multi-threaded and multi-core processor models. With performance and CPI stacks from Micro-architecture simulator, the power and energy consumption can be estimated. HotSpot [74] is employed as the thermal model. The CPI stacks and performance from micro-architecture model, power trace from power estimation model and thermal trace from thermal model are passed to reliability models discussed in section 6.1 in order to calculate the EM lifetime and SER.

After all results are obtained, they are passed to the DRL method based framework shown in Fig 6.1. Performance from micro-architecture model, average power and energy consumption from power estimation model, average temperature from thermal model and EM lifetime, SER from reliability model are passed to the learning model. Here energy is the goal to optimize and the other factors are served as constraints. With the constrains and optimization goal, we define the penalty function here. The part of a factor larger

than its constraint is multiplied with a large penalty coefficient and all these penalties and energy are used to calculate the penalty function of each action as shown in Equation 6.8.

$$PT = PT_{energy} + C \times \sum PT_{constraint} \tag{6.8}$$

Here PT is the total penalty, PT_{energy} is penalty from energy. C is the large penalty coefficient and $PT_{constraint}$ is the penalty from constraints. With these parameters, the next p-state of each core will be find. The system will run at this state for an execution period. After that the models will calculate the updated parameters and select the new optimized working state.

6.3 Numerical result and discussions

6.3.1 Experiment Setup

The proposed platform is implemented in Python 3.6.0 with with the numerical libraries (Numpy 1.11.3). Neural network is trained with tensorflow 1.3.0 [75]. Micro-architectural simulator is Sniper 6.1 [72], power estimator is McPAT 1.0.32 [73], and thermal simulator is HotSpot5.02 [74].

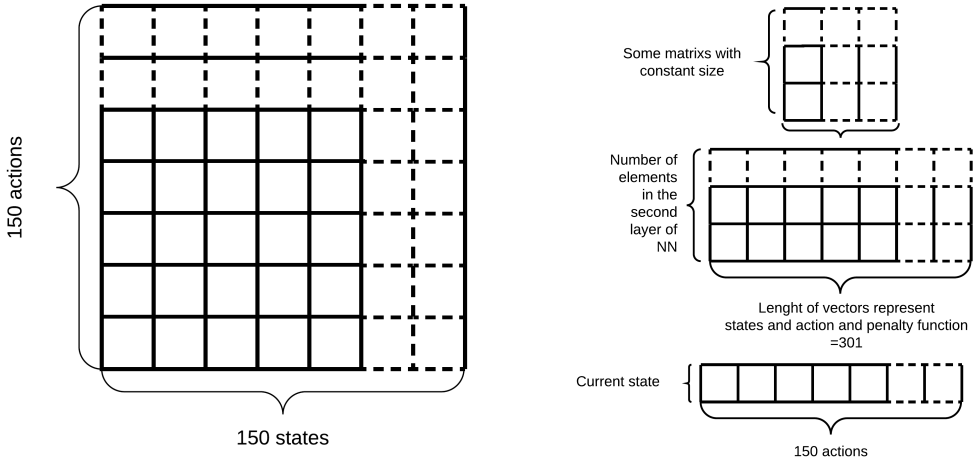
Two performance states (p-state) with the clustered DVFS [76] is chosen, which have been employed to reduce the simulation time with small solution quality degradation due to the large number of cores in our experiment. Global DVFS method, which all active cores have the same p-state, is used as the baseline to compare the energy optimization. In this work, The full power mode is (2.0GHz, 1.2V) and the low power mode is (1.0GHz,0.9V).

In the experiment, 150 states are used and there are 150 actions. The states and actions are decided by realistic processor running conditions.

6.3.2 Memory Usage

In this subsection, the memory efficiency of proposed DRL algorithm will be discussed. In the experiment 150 states are used. Q-learning needs a 150×150 size Q-table shown in Fig 6.3(a) to store Q value since in the framework the number of actions equals the number of states. In general case, the space complexity is $Q(N^2)$. Memory used of DRL method include two parts. One part is used to store the NN model and the other part is used to store the Q-value. In the memory store the NN model, since the input is represented by two vectors represent state and action, and one element for result of penalty function. The length of vector equals to the number of state and in this case the length of vector equals to 150. And we assume we have A_1 element in the layer connected with input layer. The memory taken by weight between these two layers is $(150 \times 2 + 1) \times A_1$. The general formula of this part is $(2 \times N + 1) \times A_1$ and space complexity is $O(N)$. The weight between other layers can be calculated as $A_i \times A_{i+1}$ with the element in layer i is A_i and the element in the next layer is A_{i+1}. Space complexity of this part is $O(1)$. In the Q-value part, Q-value estimated by the neural network of (s_t, a) pair needs to be stored in order to find the minimum among them. $N \times 1$ space is used to store these Q-values and in this case the size is 150. Its space complexity is $O(N)$. So as shown in Fig 6.3(b), only 150×1 space is taken. Space complexity of all these parts is $O(N)$

0.1in As can be seen, the DRL algorithm saves more memory than the traditional Q-learning method. In the experiment the Q-learning method can work since the states are



(a) (b)

Figure 6.3: (a) Memory used of q-learning (b) Memory used of DRL

not that much. However, if the p-states are increased and state-action space increase, the Q-table will be too large to store in the memory.

6.3.3 Energy Consumption

In this subsection, energy consumption of the working state selected by DRL method and Q-learning method will be compared. Since in the real application, there is limited time for state switching. So the learning process need to be finished before its deadline. In the experiment, we set the time to finish the learning is $0.008s$. This is an acceptable time for run time operation Energy consumption at the selected state with loose and tight constraint of DRL method, Q-learning and DVFS will be compared in Fig 6.4. Constraints are selected based on realistic processor running conditions.

Fig 6.4(a) shows the results using a loose bound for constraints. Power budget is $800W$, performance constraint is $40.3ms$, temperature constraint is $345K$, EM lifetime

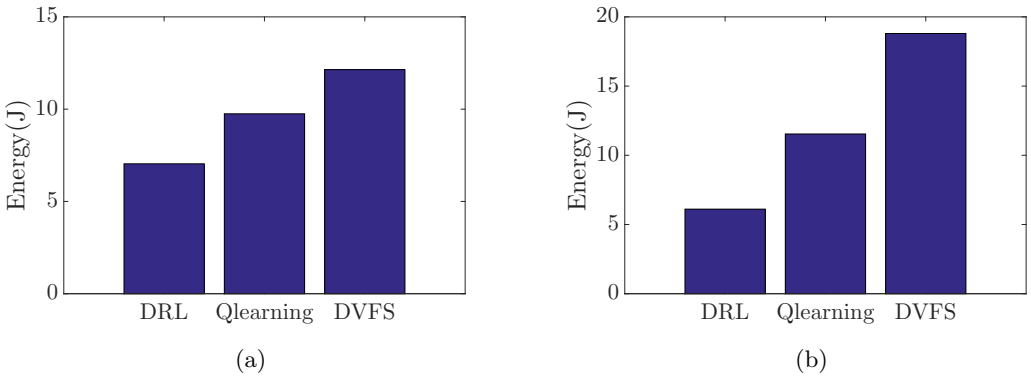


Figure 6.4: (a) Energy consumption with loose power, performance, temperature and reliability bound (b) Energy consumption with tight power, performance, temperature and reliability bound

constraint is more than 3 years and SER constraint is smaller than 0.6. As we can see, the DRL-based method can reduce energy $Energy_{reduction_ratio} = 27.79\%$ better than the Q-learning based method and 42.01% better than the DVFS method. Note that the energy reduction ratio $Energy_{r-ratio}$ between two methods is calculated by following equation:

$$Energy_{r-ratio} = \frac{E_{Q-learning} - E_{DRL}}{E_{Qlearning}} \tag{6.9}$$

Fig 6.4(b) use a tight bound for constraints. Power budget is $400W$, performance constraint is $24.1ms$, temperature constraint is $340K$, EM lifetime constraint is more than 5 years and SER constraint is smaller than 0.4. In this case, the DRL-based method can reduce energy 53.50% better than the Q-learning method and 61.29% better than the DVFS method. From the result we can see the DRL method significantly outperform the Q-learning method and the DVFS method in terms of energy reduction. Also, the energy saving of the DRL method is even more significant compared to the two methods with the tight constraint as tight constraint can help the DRL method converges faster.

6.3.4 Steps to converge

In this subsection, we compare the steps to converge of the DRL method and the Q-learning method. In the comparison, no deadline is set so both method can converge. Steps to converge of the two methods are shown in Fig. 6.5. We can see the DRL method

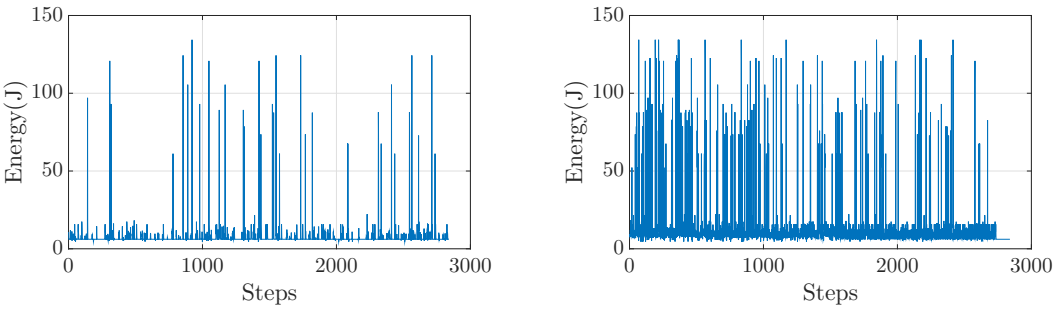


Figure 6.5: (a) Time to converge of DRL method (b) Time to converge of Q-learning method

in Fig. 6.5(a) reach steady state after few hundred iterations. That means it does not take very long time to reach the selected state. The jump after steady state is caused by random action selection, whose possibility is $1 - \varepsilon$, mentioned in 6.2.2. That can help the learning agent to find global optimizing action instead of local optimizing one. So the spikes on top does not mean the optimization is not finished. The steady state is already reached. Final state with highest possibility to appear is selected. However, the Q-learning based method keeps oscillating until around 2700 iterations as shown in Fig 6.5(b). That explains why the DRL method have better energy saving since with limited learning time, Q-learning method cannot converge to optimized state while DRL method can reach the optimized state quickly.

6.4 Summary

In this chapter, we have presented a new dynamic optimization approach with deep reinforcement learning for multi-core dark silicon processors considering both hard and soft errors. Reliability model is based on a recently proposed three phase physics-based electromigration model and an exponential soft error model considering DVFS effects. DRL which has better scalability and has lower memory complexity and computational complexity comparing with traditional reinforcement learning is applied so as to obtain cost-effective solutions for online optimization. A large class of multi-threaded applications is used as the benchmark to validate and compare the proposed dynamic reliability management methods. Experimental results on a 64-core dark silicon processor show that the proposed DRL method based framework significantly reduce memory consumed and computational time in comparison to to traditional Q-learning based method.

Chapter 7

Resource Allocation for Long-term Reliability Enhancement of Real-time Embedded Systems

In this work, we propose a novel resource allocation algorithm to improve the long-term reliability of multi-core real-time embedded systems. To the best of our knowledge, this work is the first to show the dissimilarity between utilization and reliability and propose a reliability-aware task allocation. Utilication is not the only factor related to reliability issue and other factors such as power, temperature and frequency can also hurt the reliability. We combine multiple system-level reliability models, including NBTI, HCI and EM, and apply these models to obtain a trustworthy lifetime estimation of processor cores. Our work is motivated by the fact that conventional allocation methods, such as worst-fit decreasing (WFD), consider only the amount of workload (utilization), which may have a negative

impact on lifetime. We show that the proposed algorithm yields significantly benefits in improving system-level lifetime, while guaranteeing timing correctness of all real-time tasks in the system.

7.1 Background on Reliability

Reliability is one of the most important design consideration in modern microprocessors with aggressive design strategies and decreasing feature size. It is a major challenge and limiting factor for VLSI design with increasing temperature, current, and voltage in more advanced nano-scale technology. Nowadays, with larger and more complex multi-core systems, fault tolerance in the device and system levels are demanded, so reliability models with high accuracy are required. Among all the reliability issues, NBTI, HCI and EM are the dominant effects [1, 77]. In our work, all these three reliability effects are considered in the optimization at system level. First, we will discuss how device-level models are abstracted for system-level models. Detailed reliability models used in this work will be reviewed. Then a system-level model to analyze all these reliability problems will be discussed as well as system features sensitive to reliability such as utilization and frequency.

7.1.1 Device-level to System-level Models

NBTI, HCI and EM are dominant among all reliability effects. NBTI and HCI will lead to shift of threshold voltage (V_{th}) of the impaired transistors which manifests in increasing switching and path delays [77]. EM mainly acts on a power-ground network, causes excessive IR drop, and leads to the failure of the circuit [78]. Many device-level

models are developed for these three reliability phenomena. However, in order to analyze them at the system-level, these device-level models are cumbersome and not practically applicable. To solve this problem, some system-level reliability models abstracted from the device-level models are proposed. In the following subsections, we review these system-level reliability models for NBTI, HCI and EM.

Model for Negative Bias Temperature Instability

Negative Bias Temperature Instability (NBTI) mainly affects p-type transistors (PMOS). NBTI effect consists of two phases; when the circuit is under stress, and when it is recovering. For the under stress case, logic '0' is applied to the gate of PMOS transistor and $|V_{th}|$ increase because of traps generated at the interface between gate oxide and channel. For the recovery phase, in contrast, logic '1' is applied to the same transistor which leads to decreasing of $|V_{th}|$ since some traps are filled.

Many device level models have been developed for NBTI with different parameters such as temperature (T), supply voltage (V_{dd}), and duty cycle (δ) which is the ratio between the time a transistor is under stress and total time [79]. However, in order to fit system-level optimization, a NBTI model at higher level is required.

A system level model was proposed in [80]. The model is

$$\Delta^{avg} V_{th}(t) \leq \int_0^1 A_N u(V_{dd}) \frac{(v(T_B) \cdot \delta_B \cdot \delta_e \cdot t_m)^n}{w(\delta_{B \cdot e}, T_B, t)^{2n}} d\delta_e \tag{7.1}$$

where

$$
\begin{aligned}
u =& (V_{dd} - V_{th}) \cdot exp((V_{dd} - V_{th}/E_0)) \\
v =& \xi_4 \cdot exp(-E_a/kT) \\
w =& 1 - (1 - \frac{\xi_1 + \sqrt{\xi_3 \cdot v(T) \cdot (1 - \delta(t)) \cdot t_m}}{\xi_2 + \sqrt{v(T) \cdot t}})^{\frac{1}{2n}}
\end{aligned}
\tag{7.2}
$$

Here A_n, n, E_0 and ξ_i are technology dependent constants. E_a is activation energy (positive), k is Boltzmann constant, t_m is the period between two measurements, T_B is the temperature of the transistor, δ_B is duty cycle of the block, and δ_e is effective duty cycle.

In VLSI design, the system level behavior depends on RTL level blocks and transistors in these blocks. However, with uniqueness of design and numerious of transistors in blocks, it is not realistic to get exact transistors' states for system level NBTI analysis. So in this model, the author assumes that all the transistors in the block behave similarly since RTL level description development is not available for system level analysis. Also they assume the effective duty cycle δ_e is uniformly distributed from 0 to 1.

As can be seen, the delay change is calculated for the entire block instead of a single transistor. So, average voltage shift is used here for delay estimation. This estimation might be too optimistic for a certain transistor but it is still a good estimation for system-level block. If detailed knowledge about the hardware implementation is available, partially weighted mean can be used to achieve higher accuracy.

Model for Hot Carrier Injection

Hot Carrier Injection (HCI) mainly affects n-type transistors (NMOS), where accelerated electrons inside the channel can collide with gate oxide interface and thereby create electron-hole pairs [81]. An increase in V_{th} is caused by these free electrons since they get trapped in the gate oxide layer. Based on experimental results in [82], HCI effect has sub-linear dependency on the frequency (f), runtime (t), and active factor (α), which is the ratio of the cycles in which the transistor is transitioning and the total amount of cycles. Moreover, temperature dependency is exponential for HCI. Using the same assumption used for system-level NBTI model that all the transistors in the block behave similarly. The system level model [80] is expressed as:

$$\Delta V_{th}(t) = A_H \cdot \sqrt{a_{avg,B}} \cdot U(V_{dd}) V(T_B) \cdot \sqrt{\alpha_B \cdot f \cdot t} \tag{7.3}$$

where

$$U = exp((V_{dd} - V_{th})/E_1), V(T) = exp(-E_a/kT) \tag{7.4}$$

A_H and E_1 are technology dependent constants and the activation energy E_a is positive. $a_{avg,B}$ represents the average switching activity of all gates in the micro-architectural block B. In the worst case, the average switching activity for the (near-)critical paths can be 1. So here we use 1 for $a_{avg,B}$ in order to gain maximum V_{th} shift for general application. So, if detailed information of the transistors can be obtained, $a_{avg,B}$ can be adjusted in order to get more accurate value. After we obtain the change of threshold voltage (V_{th}), the switch delay of transistor can be calculated. With the assumption we made, all

transistors inside one micro-architectural block can be represented by one single transistor T_B (*representative transistor*). Delay of the block ($\Delta^{rel} d_B$) can be estimated by relative delay if the representative transistor $\Delta^{rel} d_B = \Delta^{rel} d_{T_B}$. Thus, this delay can be calculated by a power law [83] :

$$\Delta^{rel} d_{T_B}(t) = (1 - \frac{\Delta V_{th}(t)}{V_{dd} - V_{th}(t_0)})^r - 1 \tag{7.5}$$

In this equation, V_{th} is the only variable and delay can be obtained if V_{th} is calculated. Besides HCI, NBTI delay can also be calculated using Eq. (7.5).

Model for Electromigration

Here the EM model employed is the model fro chapter 2. And for system level optimization, We assume that each core can have its own voltage (voltage island) and frequency settings, and DVFS can be done locally for each core. For the micro-architecture of EM failure we focus on power grid since it is the most vulnerable part to EM failure. A simple mesh-structured power grid is generated for each core, and each of them has its power island with the its power regulator. We assume that power grids for each core are the same and are less coupled as far as EM effects are concerned. Power for each core is estimated by a power modeling framework. For the device level power distribution, we assume that the power is evenly distributed in the power grid inside each core. This is an reasonable assumption since it is hard to consider detailed architecture inside cores. Current on each branch can be obtained from the power on each interconnect tree. Lifetime of each tree is estimated by aforementioned three phase model [9] using the corresponding current

and temperature. When the voltage drop caused by resistance increment of the power grid reaches a certain level, the core is considered as fail.

7.1.2 Unified system-level reliability model

Based on the system-level models for different reliability issues, we introduce a unified system-level reliability model used in this work based on industry standard sum-of-failure-rates (SOFR) model [63, 84]. Since detailed joint model between different reliability effects is hard to obtain, mathematical method is employed to combine these effects. Although that model may not reflect the exact joint effect of all these reliability models, it provide an efficient way to obtain the joint lifetime with different reliability models and tasks. Although some improved methods like [85] is proposed recently, SOFR is still accurate enough for most real-time systems. For multi-core systems, they will run different tasks in different frequencies. In order to consider lifetime impacts of different tasks, a system-level model is proposed in [63]:

$$MTTF = \frac{1}{(\sum_{m=1}^{n}(\Delta t_m \frac{1}{MTTF_m}))/T} \tag{7.6}$$

where $MTTF_m$ is the actual MTTF (mean time to failure) under the m-th task. Here actual MTTF means if the MTTF of a system run only one task repeatedly. For Δt_m is time period of m-th task. Assuming the chip works through n different power and frequency settings and total time T can be represneted as $T = \sum_{m=1}^{n} \Delta t_m$. Here Δt_m can be represented by

utilization. So, $MTTF$ can be defined by utilization as following:

$$MTTF = \frac{1}{(\sum_{m=1}^{n}(U_m \frac{1}{MTTF_m}))} \tag{7.7}$$

For each single $MTTF_m$, we treat lifetime as a resource and combine three different effects. Industry standard sum-of-failure-rates (SOFR) model [84] is employed as:

$$MTTF_m = \frac{1}{\frac{1}{MTTF_{NBTI}} + \frac{1}{MTTF_{HCI}} + \frac{1}{MTTF_{EM}}} \tag{7.8}$$

As can be seen, in this model, the type of tasks, period of corresponding tasks and frequencies will be the key factors of reliability analysis. The effect with the lowest lifetime dominates the $MTTF_m$. So the optimization on the critical effect can have a positive improvement in the lifetime. These three factors have the same correlation trend with frequency and utilization as well as temperature. Other important factors for $MTTF$ of each effect, such as temperature, can be obtained with architecture simulators such as Sniper and Hotspot tools. Their values also change with different frequencies and task periods for different tasks. As aforementioned, the periods of tasks can be represented by task utilization, in this work utilization is used to relax the task period in the analysis. In order to consider system-level reliability on a multi-core processor, the shortest lifetime among all the cores is employed as the lifetime for all multi-core processors [14].

7.1.3 Review of current scheduling techniques considering reliability

Scheduling techniques are employed in embedded systems in order to address long-term reliability challenges [17,18]. A simulated annealing-based task allocation and scheduling strategy is proposed in [17]. This strategy can extend lifetime of platform-based MPSoC design significantly with performance constrains. A cost function is defined in that work which is:

$$Cost = \mu \cdot 1_{\{\exists i: e_i > d_i\}} - MTTF \tag{7.9}$$

where μ is a significant large number, and 1. is the indicator function which equals to 1 if deadline is missed and equal to 0 if the scheduler can meet the deadline. As can be seen, the cost function can decrease with higher MTTF but will be significantly higher if the deadline missed. Simulated annealing can obtain the lowest cost, which means the longest MTTF can be achieved with a performance constraint. However, they use only simple general model for MTTF which is not realistic and do not have physics meaning.

Physics-based EM model is employed in [18] as the reliability model for lifetime task optimization technique. This model is a two phase model proposed in [7] and only nucleation phase and growth phase are considered. DVFS is employed first for each task and corresponding MTTFs of all the performance states (p-state) are calculated. Best p-state with maximum lifetime for each task with timing constraint is obtained by equation (7.10).

$$\text{Max } Lifetime(m) = \frac{1}{(\sum_{n}^{i=1}(\Delta t_i \frac{1}{MTTF_i} \frac{T_{hyper}}{T_i}))/T_{hyper}}$$

Subject to:

$$r_i = \Delta t_i + \sum_{k \in hp(i)} \left\lceil \frac{r_i}{T_k} \right\rceil \Delta t_k \leq T_i$$

$$MTTF_{i,l} \leq MTTF_i \leq MTTF_{i,u} \tag{7.10}$$

where m is the variable vector for MTTF, i is the task id, Δt_i is the execution time for task i, n is the total number of task, $MTTF_i$ is the segment MTTF for task i, r_i is the response time of task i, $hp(i)$ is the task set containing the higher priority tasks than the current task i. T_i is the task period of task i, which is a deadline. T_{hyper} is the hyper period for all tasks. At the single-rate T_{hyper} is equal to T_i. T_{total} is the total execution time of all tasks. $MTTF_{i,l}$ is the minimum bound of MTTF for task i and $MTTF_{i,u}$ is the maximum bound of MTTF for task i.

Response surface method (RSM) and adaptive simulated annealing (ASA) are used optimize constrained non-linear formula in Eq. (7.10) in order to find optimized p-state. However, this method assumes a uniprocessor real-time system and cannot consider the multi-core systems which is generally used in the application. Also, only EM is considered in this work and other important reliability issues such as NBTI and HCI are ignored.

Recent work [19] proposes an optimization method for system-level reliability in multi-core systems. Their goal is to minimize the deviation of cores' utilization from a set point in each time interval with certain constrains:

$$min \sum_{pi \in M} (U^s - U_i(k))^2 \tag{7.11}$$

Although four reliability factors (EM, TDDB, SM and TC) are considered in [19], their primary focus is on temperature since EM, TDDB, and SM are strongly dependent on it. However, besides temperature, there are also many other momentous factors such as power and voltage which are not considered in their work. These factors can cause lifetime

variation among tasks. So, balanced utilization does not necessarily lead to optimized MTTF due to lifetime variation among tasks. We address such limitations in this paper.

7.2 System Model

The real-time system considered in this work is equipped with a single-chip multi-core processor. The processor has N_P identical cores, each of which can be configured with different operating frequencies and voltage levels. This assumption is supported by other existing works [86] and existing processors like ARM Cortex A75 also support per-core DVFS [87]. The system runs an operating system that is capable of scheduling n tasks based on task priorities. We focus on *partitioned preemptive earliest-deadline-first scheduling (P-EDF)* due to many benefits: (i) partitioned scheduling does not introduce task migration costs, thereby widely used in safety-critical embedded real-time systems, and (ii) on each core, EDF is known to be optimal to schedule tasks with timing constraints. Under P-EDF, each task is statically assigned to a single CPU core. Any arbitrary tie-breaking rule can be used to determine the priorities of tasks with the same deadline, e.g., task indices.

We consider the periodic real-time task model [88], which is a widely used model in the literature. Each task τ_i is represented by the following parameters:

$$\tau_i := (C_i^f, T_i, D_i)$$

- C_i^f: the worst-case execution time (WCET) of a single instance (called *job*) of the task τ_i, when it executes on a core running at f Hz of frequency

- T_i: the period of τ_i
- D_i: the relative deadline of τ_i ($D_i \leq T_i$)

Note that for notational simplicity, C_i may be used instead of C_i^f in the rest of the paper, if the frequency of τ_i's core is assumed to be known. The utilization of a task τ_i is defined as $U_i = C_i/T_i$. Without loss of generality, we assume implicit deadline for all tasks, i.e., $T_i = D_i$. If the deadline of a task τ_i can be satisfied under any circumstance, τ_i is said to be *schedulable*. The schedulability of tasks under P-EDF can be checked on a per-core basis. Tasks on a core p are schedulable if the following condition satisfies:

$$\sum_{\forall \tau_i : \tau_i \in \Gamma_p} C_i/T_i \leq 1 \tag{7.12}$$

where Γ_p is the set of tasks allocated on the core p.

As our main concern is system-level reliability, we assume that task execution time is not affected by contention in multi-core shared hardware resources, such as caches and memory buses. Details on accounting for the delay from such resources can be found in [89–95]. Note that our proposed work can be easily integrated with them by incorporating the blocking terms obtained by those approaches into Eq. (7.12).

7.3 Reliability-aware Real-Time Resource Allocation

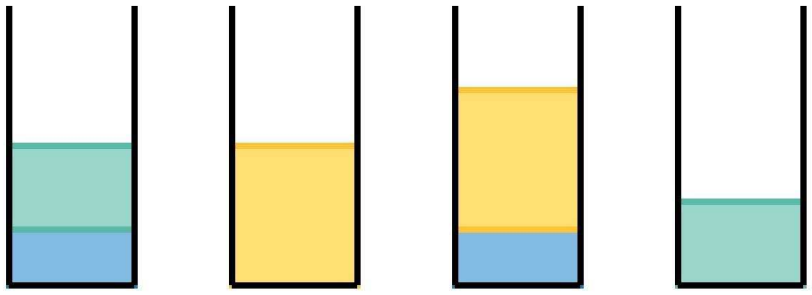
In this section, we propose our resource allocation algorithm that improves the long-term reliability of multi-core real-time systems. We first discuss the challenges observed from motivational experiments and then provide details on the proposed algorithm.

7.3.1 Motivational experiments and key observations

In a multi-core system, the problem of finding an optimal allocation of tasks to cores is typically modeled as a *bin-packing problem* that is known to be NP-complete [96]. As the allocation may need to be changed at runtime even in real-time embedded systems, e.g., mode changes for different missions, bin-packing heuristics are widely used to find a near-optimal solution , such as worst-fit decreasing (WFD) best-fit decreasing (BFD) and first-fit decreasing (FFD). These heuristics first sort tasks in decreasing order of their utilization so that large tasks can be allocated first, which helps improve the success rate or reduce the number of required cores in general. WFD puts tasks in the emptiest core, BFD puts tasks in the fullest core which they fit and FFD puts tasks in the core opened earliest which they fit.

System-level reliability is largely affected by task allocation. As it is good to balance the wear-out of cores, it is seemingly intuitive to evenly distribute the load across cores, which WFD does. However, we find that this approach does not yield long system-level lifetime due to a huge lifetime variation among tasks. For instance, tasks with high utilization do not necessarily have a short lifetime. If such tasks are allocated by WFD, the utilization of cores will be balanced but not the lifetime. Fig. 7.1 illustrates an example of 3 tasks allocated to 2 cores, where each bin represents a core. The MTTF and utilization of individual tasks are shown in Table 7.1. Note that the $MTTF_i$ in Table 7.1 is calculated by Eq. 7.7. As can be seen in Fig. 7.1(a), the utilization is balanced (0.5 for each core), but the system-level lifetime is only 2.68 years based on Eq. 7.7. In Fig. 7.1(b), the utilization of the first core is 0.7 and that of the second core is 0.3. Although it is unbalanced in terms

of utilization, the system-level lifetime is 3.9 years based on Eq. 7.6, which is 45% longer than the prior case.



(a) (b)

Figure 7.1: (a) Balanced utilization with short lifetime. (b) Unbalanced utilization with long lifetime

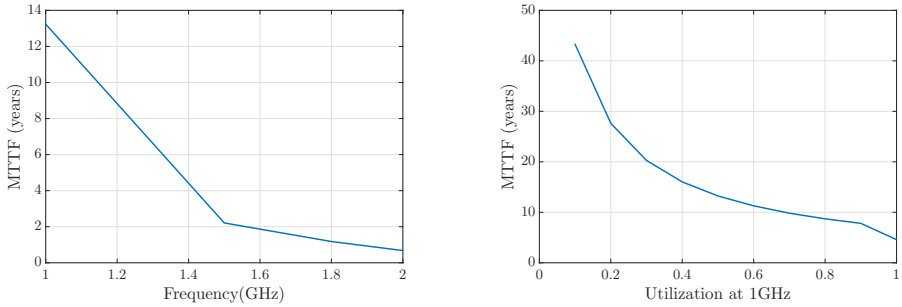
From the above example, we can observe that balanced utilization is not enough for lifetime optimization. Task allocation should consider the fact that each task has a different impact on lifetime regardless of its utilization; e.g., tasks with low (or high) utilization may have short (or long) lifetime.

For the same type of tasks, the operating frequency of assigned cores and task utilization play a dominant role in lifetime. Fig. 7.2(a) shows the lifetime (MTTF) of a task under different frequency. Note that C_i^f (WCET) is different with different frequency since the utilization is fixed. As can be seen, MTTF decreases rapidly with increasing frequency.This trend is also observed in [97]. In such a case, the core should run at the lowest feasible frequency where all the tasks of that core remain schedulable, in order to achieve the longest possible lifetime. For a task executing at a fixed frequency, as shown in Fig. 7.2(b), MTTF decreases with increasing utilization. In this case, C_i^f is different with different utilization with a fixed frequency. The sudden change at 1.5GHz in Fig. 7.2(a) and

Table 7.1: Example task parameters for Fig. 7.1

Task τ_i	**Utilization** U_i	**MTTF** $MTTF_i$
τ_1 (yellow)	0.5	39.8
τ_2 (green)	0.3	3.9
τ_3 (blue)	0.2	8.6

at 0.9 utilization in Fig. 7.2(b) are caused by change of domination of different reliability factors. This will be discussed in detail in section7.4.1.



(a) (b)

Figure 7.2: (a) MTTF at different frequencies for task utilization of 0.5. (b) MTTF at 1 GHz for different task utilizations

With the aforementioned problems, a new algorithm is required for task allocation and CPU frequency selection. Different impact sources on lifetime, such as tasks and running frequency and utilization of cores, need to be taken into consideration for lifetime optimization.

7.3.2 Proposed algorithm

Motivated by the observations in the previous subsection, we proposed a reliability-aware algorithm. Our goal is to maximize system-level lifetime while guaranteeing the schedulability of all tasks in the system. To achieve this goal, our algorithm integrates lifetime estimation and frequency selection into task allocation. As the problem of task

allocation itself is NP-complete [96], we develop a heuristic so that it can be used at runtime when re-allocation is needed.

Algorithm 1 Reliability-aware Resource Allocation

Input: Γ: a taskset (all tasks in the system), $\mathbb{P}$: a set of all available CPU cores in the system, $\mathbb{F}$: a set of available frequencies
Output: Success or failure

1: /* Get number of cores and nominal frequency */
2: $N_P \leftarrow |\mathbb{P}|$
3: $f_{nom} \leftarrow \min_{f \in \mathbb{F}} f$ s.t. $\sum_{\tau_i \in \Gamma} C_i^f / T_i \leq N_P$
4: **if** there is no feasible f_{nom} **then** failure
5: **end if**
6: /* Divide tasks into two subsets: Γ_{large} and Γ_{small} */
7: $\Gamma_{large} \leftarrow \bigcup_{\forall \tau_i \in \Gamma \wedge U_i \geq 0.5} \tau_i$
8: $\Gamma_{small} \leftarrow \Gamma \setminus \Gamma_{large}$
9: /* Allocate large tasks first */
10: **for all** $\tau_i \in \Gamma_{large}$ in inc. order of lifetime at f_{nom} **do**
11: lifetime$_{max} \leftarrow 0$
12: **for all** $p_j \in \mathbb{P}$ **do**
13: $f_{p_j} \leftarrow \min_{f \in \mathbb{F}} f$ s.t. $\sum_{\tau_k \in \Gamma_{p_j} \cup \{\tau_i\}} C_k^f / T_k \leq 1$
14: **if** there is no feasible f_{p_j} **then**
15: **continue** /* τ_i is unschedulable on p_j */
16: **end if**
17: lifetime$_{p_j} \leftarrow$ Lifetime of $\Gamma_{p_j} \cup \{\tau_i\}$ at f_{p_j} by Eq. (7.8)
18: **if** lifetime$_{p_j} >$ lifetime$_{max}$ **then**
19: lifetime$_{max} \leftarrow$ lifetime$_{p_j}$
20: $p^* \leftarrow p_j$
21: **end if**
22: **end for**
23: **if** lifetime$_{max} = 0$ **then** failure /* there is no core to schedule τ_i
24: **else**
25: $\Gamma_{p^*} \leftarrow \Gamma_{p^*} \cup \{\tau_i\}$
26: **end if**
27: **end for**
28: Do the steps from lines 10 to 27 for Γ_{small} success

Our proposed algorithm is given in Algorithm 1. It takes three input parameters: Γ is the taskset to be considered, $\mathbb{P}$ is the set of cores in the system, and $\mathbb{F}$ is the set of available operating frequencies of $\mathbb{P}$. The algorithm returns *success* if all tasks are schedulable, and *failure* otherwise. Task allocation and frequency selection results can be easily obtained by checking the corresponding local variables before completion.

The algorithm first computes the nominal frequency f_{nom} for a given taskset (line 3), which is the minimum frequency required to schedule tasks in Γ, i.e., total utilization does not exceed N_P. Hence, f_{nom} can be said to be the optimal frequency for a given taskset. Here the optimal frequency means the lowest possible frequency to run the taskset which will lead to longest $MTTF$. However, it is possible that the taskset cannot be partitioned and f_{nom} need to be increased to minimum feasible frequency f_{p_j} in the following part of the algorithm. The algorithm immediately returns tasks which are unschedulable by any frequency in $\mathbb{F}$. It then divides Γ into two subsets: Γ_{large} and Γ_{small} (lines 7 to 8). If the utilization of a task is larger than 0.5 at f_{nom}, it is classified as a *large* task; otherwise, it is a *small* task. The reason for dividing tasks into two groups is to allocate large tasks first as they have a higher impact on allocation than small tasks. The threshold of 0.5 is widely used to distinguish large tasks in the literature because no more than one task with 0.5 utilization can be allocated to the same core.

Once tasks are divided, the algorithm allocates tasks in Γ_{large} in increasing order of their lifetime computed at f_{nom} (line 10). Hence, tasks with shorter lifetime are allocated first as they have a higher impact on the system-level reliability. Note that this ordering does not give a significant disadvantage to the allocation performance of our algorithm, compared to WFD, BFD and FFD, because the algorithm allocates large tasks separately from small tasks. For each task τ_i, the algorithm computes the expected lifetime of each core p_j when τ_i is allocated to it. Specifically, in line 13, it finds the minimum feasible frequency f_{p_j} for the core p_j such that all tasks on p_j and τ_i ($\Gamma_{p_j} \cup \{\tau_i\}$) are schedulable at f_{p_j} under EDF (see Eq. (7.12)). It then computes the lifetime of $\Gamma_{p_j} \cup \{\tau_i\}$ at f_{p_j} using Eq. (7.8),

and takes that value as the expected lifetime of p_j (lifetime$_{p_j}$). If it fails to compute the lifetime of any core at all, this means there is no core to schedule τ_i and thus the algorithm returns failure (line 23). Otherwise, the algorithm allocates τ_i to the core with the longest expected lifetime among all cores. Once the allocation of large tasks is done, the algorithm performs the same steps for small tasks (line 28). Finally, the algorithm returns success. The time complexity of the proposed algorithm is $O(n^2 \cdot N_P \cdot |\mathbb{F}|)$, where n is the number of tasks and $|\mathbb{F}|$ is the number of frequency levels.

7.3.3 Applicability to sporadic tasks and fixed-priority scheduling

It is worth noting that our proposed work can also support the sporadic task model [98] where each task has the minimum inter-arrival time between any two consecutive instances (called *jobs*) of the task. In this case, some jobs of a task may have inter-arrival times longer than the task's minimum inter-arrival time. However, larger inter-arrival times will benefit, or increase, the MTTF. The reasons are listed in the following. (i) In the NBTI model given by Eq. (7.1), since the block duty cycle δ_B will be smaller when the inter-arrival time is longer, the corresponding lifetime of NBTI will be increased. (ii) The same phenomenon can be observed in the HCI model in Eq. (7.3). With longer inter-arrival times, active factor (α) which represents the ratio of the transistor transitioning cycles and total amount of cycles will be smaller which results to increase of lifetime. (iii) EM lifetime also increase since the idle times are increased with longer inter-arrival times. Therefore, the estimated MTTF using the minimum arrival time will be the worst case MTTF because it will have the longest δ_B, largest α and the shortest idle time for three reliability factors. Although this paper assumes partitioned EDF scheduling, our algorithm

given in Algorithm 1 can be easily extended for systems using fixed-priority scheduling on each core. Basically, the line 13 of Algorithm 1 finds the minimum feasible core frequency by using the schedulability condition under EDF given in Eq. 7.12. This condition can be easily changed to the one for fixed-priority scheduling, such as the Rate Monotonic utilization bound [88] and the iterative response-time test [99]. The other parts of the algorithm do not need any modification as they will be unaffected by the change of the schedulability condition.

7.4 Evaluation

In this section, we will evaluate the effectiveness of our proposed resource allocation algorithm with randomly-generated tasksets. The performance metric we are interested in is the system-level MTTF (also called *lifetime*) while all tasks in the system meet their real-time constraints.

7.4.1 Taskset Generation

A modified architecture simulator (Sniper 6.1 [22]) with SPLASH-2 multi-threaded application benchmarks [71] at different DVFS levels is used to capture the reliability characteristics of realistic workloads and to use them in real-time taskset generation. Power is obtained by the power simulator (McPAT 1.0.32) [73] and temperature is obtained by temperature simulator (HotSpot 5.02) [74]. In this work, 45nm-based simulation framework is employed. Among a total of eleven SPLASH-2 benchmarks, we choose the following seven benchmarks: `barnes`, `cholesky`, `fft`, `fmm`, `radiosity`, `radix`, and `raytrace`, which are

chosen to represent a combination of long and short lifetime tasks. The simulated processor architecture has 4 power modes: (2 GHz, 1.2 V), (1.8 GHz, 1.1 V), (1.5 GHz, 1.0 V), and (1.0 GHz, 0.9 V). We assume core DVFS sets with voltage and frequency scaling proportionally, with frequencies in the extracted range (1.0 GHz-2.0GHz) from the cpufreq governors [100], which is based on Enhanced Intel Speedstep Technology [101], and with voltages in 4 equally-spaced voltages in the rage 0.9 V-1.2V from 45-nm technology.

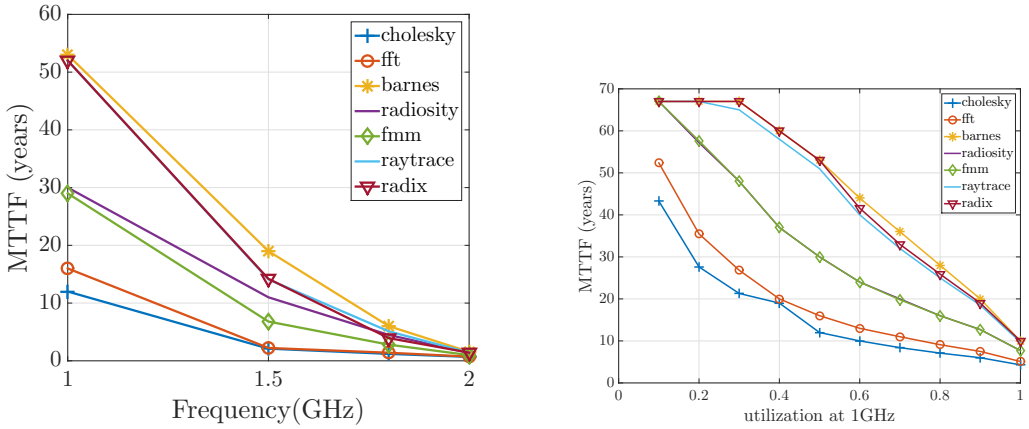
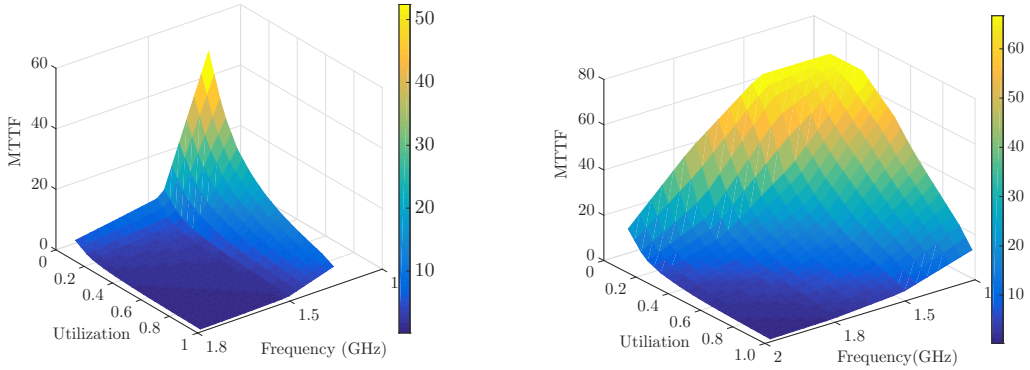


Figure 7.3: (a) MTTF for all workloads at different frequencies with 0.5 utilization. (b) MTTF for all workloads at 1 GHz with different utilizations

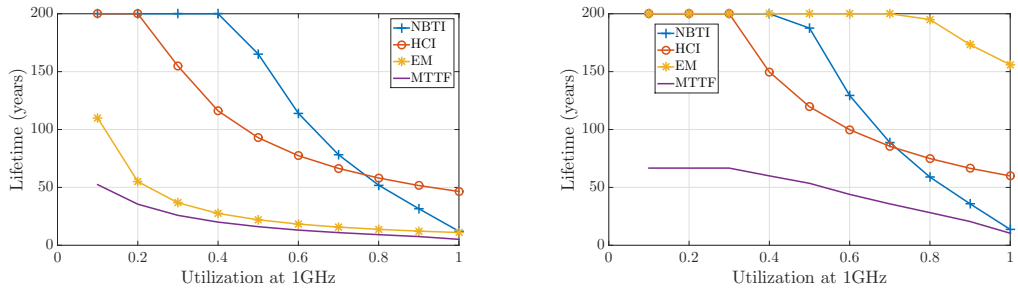
Fig. 7.3 shows the MTTFs of the SPLASH-2 benchmarks as the operating frequency and the utilization of each benchmark change. As can be seen, `barnes`, `radix` and `raytrace` have longer lifetime than the other four benchmarks. This is because these benchmarks require lower power as well as lower temperature during their run time comparing with other tasks running at same frequencies.

The MTTFs of the benchmarks are computed considering the three reliability effects discussed in Section 7.2. Each benchmark used has a different dominant effect on



(a) (b)

Figure 7.4: (a) Lifetime on different frequency and utilization of benchmark `fft`. (b) Lifetime on different frequency and utilization of benchmark `barnes`.



(a) (b)

Figure 7.5: (a) Lifetime of different effects for benchmark `fft` at 1GHz. (b) Lifetime of different effects for benchmark `barnes` at 1GHz.

its MTTF. As an example, we compare in Fig. 7.5 the impact of the three reliability effects on the lifetime of the two benchmark, `fft` and `barnes`. The lifetime of `fft` is dominated by EM, and NBTI causes a higher impact than HCI when the utilization is high. On the other hand, the lifetime of `barnes` is dominated by HCI if the utilization is lower than 70%, and by NBTI otherwise. That means that, depending on the type, operating frequency and utilization of a benchmark, the dominant effect on MTTF can be different. Based on these results, we have classified the benchmarks into two groups: the one is for high-lifetime

tasks and the other is for low-lifetime tasks. The impact of the percentage of high- and low-lifetime tasks on the system-level MTTF will be studied in the next subsection.

For each experimental setting, we use 1,000 randomly-generated tasksets. Table 7.2 shows the base parameters used for taskset generation. Systems with four to sixty four cores are considered ($N_P = 4, 8, 12, 16, 32, 64$). For each taskset, n tasks are generated where n is randomly chosen from $[N_P, 4N_P]$. The type of each task is randomly picked from the aforementioned seven SPLASH-2 benchmarks and is used to determine the reliability characteristics of that task. The period of a task τ_i is chosen from $[100, 500]$ ms. The utilization of τ_i at 2 GHz (U_i^{2G}) is randomly chosen from $[0.05, 0.5]$, and the worst-case execution time (WCET) of τ_i at 2 GHz (C_i^{2G}) is obtained by $U_i^{2G} \cdot T_i$. For simplicity, we assume that task utilization is linear to core frequency, e.g., the range of task utilization at 1 GHz is $[0.1, 1]$.

7.4.2 Experimental Results

For comparison with our proposed method, we consider variants of the worst-fit decreasing (WFD) best-fit decreasing (BFD) and first-fit decreasing (FFD) heuristics. Each of these heuristics first allocates tasks to cores assuming all the cores run at the maximum frequency (2 GHz). As aforementioned, minimum feasible frequency need to be found in order to find longest system MTTF, it then reduces the frequency of each core p_j as much as possible while all tasks allocated to p_j remain schedulable, i.e., minimize $f : f \in \{2G, 1.8G, 1.5G, 1G\}$ s.t. $\sum_{\tau_i \in \Gamma_{p_j}} C_i^f / T_i \leq 1$. Finally, the MTTF (lifetime) of each core is computed and the shortest MTTF is taken as the system-level MTTF. The three heuristics combined with this approach serve as the baselines in our experiments since the existing

Table 7.2: Base parameters for taskset generation

Parameters	**Values**
Number of cores (N_P)	4, 8, 12, 16, 32, 64
Number of tasks (n)	$[N_P, 4N_P]$
Task type	One of SPLASH-2 benchmarks
Task util. at 2 GHz (U_i^{2G})	[0.05, 0.5]
Task period ($T_i = D_i$)	[100, 500] ms
Task WCET at f Hz (C_i^f)	$U_i^f \cdot T_i$

task allocation approaches developed for reliability, such as [102–104], focus on only the utilization of cores during allocation and thus share the same idea as these heuristics.

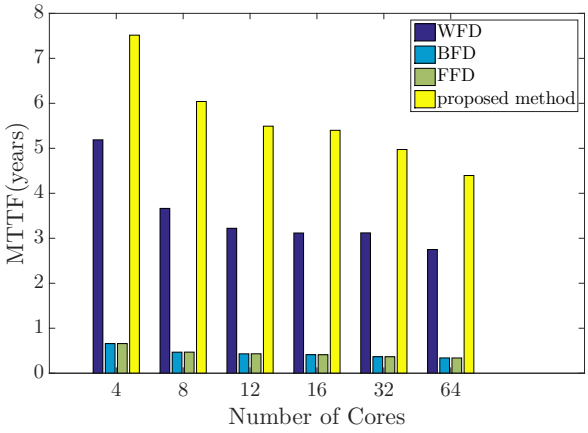


Figure 7.6: MTTF for a different number of cores (N_P) with n tasks randomly chosen from $[N_P, 4N_P]$

Fig. 7.6 shows the MTTF of a system with a different number of cores (N_P). For each N_P from $\{4, 8, 12, 16, 32, 64\}$, all other parameters are randomly chosen from the ranges given in Table 7.2. As can be seen, MTTF decreases with an increment of N_P. With more cores, more tasks can be scheduled so that cores run at a higher frequency and the utilization of them increases. The proposed method gives the highest improvement over the other methods when the system has 12 cores. The maximum lifetime improvement by our method is 70.45% compared to WFD and 1219.67% compared to BFD and FFD.

The reason for the bad performance of BFD and FFD is that they lead to imbalanced core utilization, thereby requiring higher frequency.

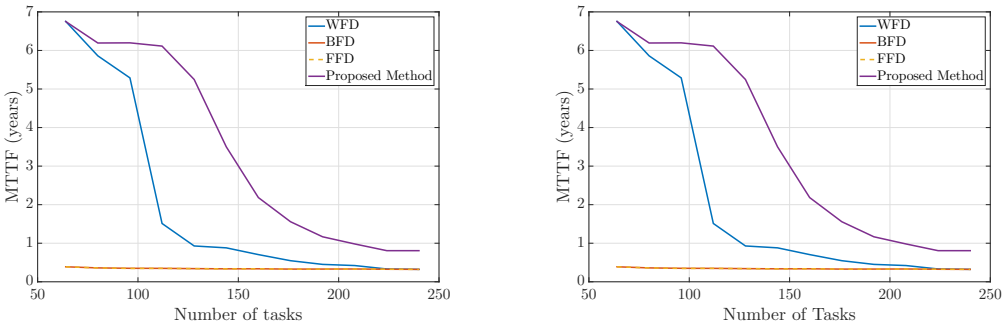


Figure 7.7: (a) MTTF for a different number of tasks from 8 to 32 with 8 cores. (b)MTTF for a different number of tasks from 64 to 256 with 64 cores.

Fig. 7.7 depicts the MTTF with respect to the number of tasks in the system. N_p is set to 8 and 64 in this experiment, and the number of tasks is chosen form $[N_P, 4N_p]$ with a step equals to 2. We can see that the lifetime decreases as the number of tasks increases. With more tasks, our proposed methods give a higher percentage of lifetime improvement over the other methods. For instance, when there are 18 tasks, there are 182.7% increment in lifetime compared to WFD and 1404.0% increment in lifetime compared to BFD and FFD with 8 core case. For 64 core case, when there are 128 tasks there are 464% increment in lifetime compared to WFD and 1438.7% increment in lifetime compared to BFD and FFD. However, the absolute value of MTTF becomes lower as the number of tasks increases because core frequencies will need to increase to meet task deadlines.

Fig. 7.8 shows the MTTF for different percentage of low-MTTF tasks (task type={`cholesky`,`fft`, `fmm`,`raytrace`}). N_P is set to 8 and 64 and corresponding

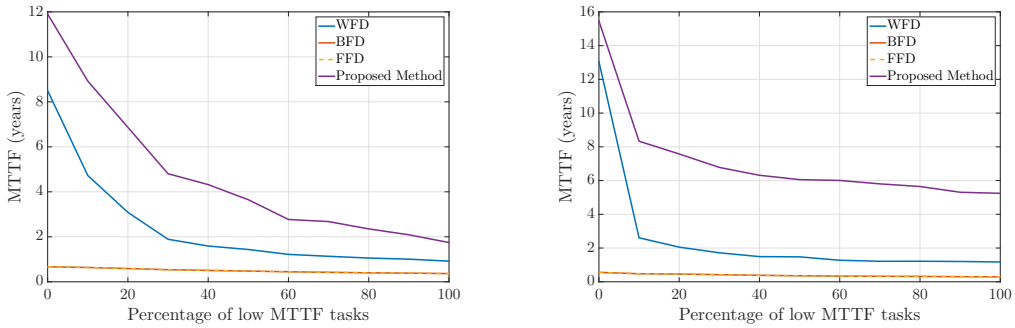


Figure 7.8: (a) MTTF for different percentage of low-MTTF tasks with 8 cores and 20 tasks. (b) MTTF for different percentage of low-MTTF tasks with 64 cores and 160 tasks.

numbers of tasks are set to 20 and 160. For each taskset with 20 tasks, we randomly choose the percentage of low-MTTF tasks in the taskset from the range of $[0, 100]$. It can be seen in the figure that lifetime reduces with higher percentage of low-MTTF tasks. When there are 50% of low-MTTF tasks in a taskset, the proposed method achieves the largest improvement in lifetime over the other methods, 171.7% over WFD and 755.6% over BFD and FFD for 8 core case and for 64 core case, the proposed method achieves the largest improvement in lifetime over the other methods, 378.1% over WFD and 1696% over BFD and FFD when there are 80% of low-MTTF tasks in a taskset.

As aforementioned, proposed method has better lifetime comparing with other method since lifetime is balanced across all CPU cores. Table 7.3 shows the statistics of lifetime of a system with 64 cores. As shown in the table, with BFD and FFD method, some cores are immortal while the minimum lifetimes are only 0.384 years. And proposed method has higher minimum lifetime and smaller variance comparing with WFD. So we can see proposed method achieve balanced lifetime across all cores.

Table 7.3: statistics of lifetime of a system with 64 cores

Lifetime (years) \ Method	WFD	BFD	FFD	Proposed Method
Max	62.5	Immortal	Immortal	26.6
Min	5.99	0.384	0.384	7.56
Average	20.0	-	-	15.8
Variance	136	-	-	10.8

Based on these results, we conclude that our proposed method yields a significant benefit in the system-level lifetime while satisfying the timing constraints of real-time tasks. In this work, we propose a novel resource allocation algorithm to improve the long-term reliability of multi-core real-time embedded systems. To the best of our knowledge, this work is the first to show the dissimilarity between utilization and reliability and propose a reliability-aware task allocation. Utilication is not the only factor related to reliability issue and other factors such as power, temperature and frequency can also hurt the reliability. We combine multiple system-level reliability models, including NBTI, HCI and EM, and apply these models to obtain a trustworthy lifetime estimation of processor cores. Our work is motivated by the fact that conventional allocation methods, such as worst-fit decreasing (WFD), consider only the amount of workload (utilization), which may have a negative impact on lifetime. We show that the proposed algorithm yields significantly benefits in improving system-level lifetime, while guaranteeing timing correctness of all real-time tasks in the system.

7.5 Summary

In this chapter, we have presented a long-term reliability-aware resource allocation algorithm for multi-core real-time embedded systems. Our work considers system-level

models for long-term reliability effects including EM, NBTI and HCI. The lifetime obtained with these models is treated as resource to be controlled. Our proposed resource allocation algorithm uses this information to improve the lifetime of a system under partitioned EDF scheduling. Unlike the conventional allocation methods for partitioned scheduling, such as WFD, BFD, and FFD, which are agnostic to reliability, our algorithm achieves balanced lifetime across all CPU cores. Experimental results show that the proposed algorithm can achieve significant improvement in MTTF, which means that our algorithm substantially outperforms the conventional scheduling methods in terms of long-term reliability. In the future , reliability on single core after task allocation will also be studied.

Chapter 8

Conclusion

Physics level and System level reliability has been getting worse with more aggressive design strategy. It is expected that reliability will be one of the most serious issue for VLSI design, especially for advanced sub 7nm technology node. Both physics level reliability for interconnects on chip and system level reliability for multi core systems need to be held in high regard. In this dissertation, we propose physics level EM model for VLSI circuits and system level EM optimization for multi-core systems. In this chapter, the main contributions are summarized.

8.1 Voltage based EM immortality check

In chapter 3 we have proposed a EM immortality check for general multi-segment interconnect wires. The new method estimates the EM-induced steady-state stress in general multi-segment copper interconnect wires based on a novel parameter, *Critical EM Voltage,* $V_{Crit,EM}$. The proposed method, called voltage-based EM or *VBEM* method, mit-

igates the problem of current-density-based EM criteria, which can only be applied to a single wire. The new VBEM method can naturally comprehend the impact of the topology of the wire structure on EM-induced stress. As a result, this new VBEM analysis method is very amenable to addressing EM violations, as it brings new optimization capabilities to the physical design flow. The VBEM stress estimation method is based on the fundamental steady-state stress equations. This approach avoids computationally-intensive numerical methods and can be implemented in CAD tools very easily. We also study the impact of current crowding in practical interconnect wires on the estimated steady-state stress, which are shown to be not significant if the length of the wire is much greater than its width. An extension of the VBEM method to consider the significant current crowding effects is also shown and additionally, we analyze mesh-structured interconnect wires and demonstrate that the proposed VBEM method is correct and accurate on such structures.

8.2 Saturation volume based EM immortality check

In chapter 4 we have proposed a new formula of the void's saturation volume for general multi-segment interconnect wires. The new formula is based on the fundamental atom conservation at the steady-state condition of void growth phases. The new void saturation estimation formula agrees with the existing single segment wire saturation void volume formula and is an natural extension of the single segment case to general multi-segment wires. In addition, we consider impacts of the void volume on final stress distributions of the wire to further improve the accuracy of the proposed formula. The proposed saturation void volume estimation can be applied for fast EM immortality check for nucleated wires,

which were considered to be failed wires in the past. Combined with the recently proposed EM immortality check for nucleation, we propose a new EM immortality check algorithm, which considers both void nucleation and void growth for the first time and thus overcoming the conservativeness of the existing EM immortality check method. Our numerical results show that the proposed formula agrees well with a published work for two-segment cases, which are supported by experimental data. The formula is also validated by the recently proposed physics-based 3D finite element (FEM) analysis tool for general multi-segment interconnect wires.

8.3 Coupled Electronic and Stress Simulation

In this chapter 5 we have proposed *EMSpice*, which is a new full-chip EM failure analysis tool for physics-based coupled EM and electronic analysis. *EMSpice* takes power grid netlists from Synopsys ICC flow, and outputs the failed EM wires and their resistance changes and resulting IR drops of the power grids over the given aging time. Hydrostatic stress and electronic current/voltage in a power grid network can be considered simultaneously. *EMSpice* simulator employs a finite difference time domain (FDTD) solver for stress analysis for every interconnect tree for both nucleation and post-voiding phases. *EMSpice* simulator also couples the EM analysis with IR drop analysis at time domain. Thus the solver can consider the interplay among stress, void growth, resistance change and IR drop in a single simulation framework. Furthermore, *EMSpice* simulator considers both recently proposed nucleation phase immortality and incubation phase immortality for the first time to remove immortal interconnect trees from EM analysis.

8.4 Deep Reinforcement based reliability management

In chapter 6 we have proposed a new deep reinforcement learning optimization frame work considering both hard and soft errors for dark silicon processors. We formulate our DRM problem as minimizing the energy consumption subject to the reliability include hard and soft errors, power, performance and thermal constraints. The hard error model is based on a recently proposed physics based EM model. Hard and soft errors are obtained by recently proposed reliability models. Power, performance are acquired by X86 based micro-architecture model and thermal information is given by temperature simulators. The control operation we use dynamic voltage and frequency scaling (DVFS) and ON/OFF pulsing action. DRL method is used to select the most cost effective operation. Furthermore, the DRL-based method also shows much smaller space complexity (linear versus the quadratic) and less steps to converge than the Q-learning based method as well.

8.5 Resource allocation based reliability management

In chapter 7 we have proposed a novel resource allocation algorithm to improve the long-term reliability of multi-core real-time embedded systems. To the best of our knowledge, this work is the first to show the dissimilarity between utilization and reliability and propose a reliability-aware task allocation. Utilization is not the only factor related to reliability issue and other factors such as power, temperature and frequency can also hurt the reliability. We combine multiple system-level reliability models, including NBTI, HCI and EM, and apply these models to obtain a trustworthy lifetime estimation of processor cores. We show that the proposed algorithm yields significantly benefits in improving system-level

lifetime over conventional methods, while guaranteeing timing correctness of all real-time tasks in the system.

Bibliography

[1] "International Technology Roadmap for Semiconductors 2.0 (ITRS 2.0)," 2015. http://www.itrs2.net/itrs-reports.html.

[2] S. R. Nassif, "Power Grid Analysis Benchmarks," in *Proc. Asia South Pacific Design Automation Conf. (ASPDAC)*, pp. 376–381, Mar. 2008.

[3] B. Bailey, "Thermally Challenged," *Semiconductor Engineering*, pp. 1–8, Dec. 2013.

[4] J. R. Black, "Electromigration-A Brief Survey and Some Recent Results," *IEEE Trans. on Electron Devices*, vol. 16, no. 4, pp. 338–347, 1969.

[5] I. A. Blech, "Electromigration in thin aluminum films on titanium nitride," *Journal of Applied Physics*, vol. 47, no. 4, pp. 1203–1208, 1976.

[6] M. A. Korhonen, P. Bo/rgesen, K. N. Tu, and C.-Y. Li, "Stress evolution due to electromigration in confined metal lines," *Journal of Applied Physics*, vol. 73, no. 8, pp. 3790–3799, 1993.

[7] X. Huang, T. Yu, V. Sukharev, and S. X.-D. Tan, "Physics-based electromigration assessment for power grid networks," in *Proc. Design Automation Conf. (DAC)*, pp. 1–6, IEEE, June 2014.

[8] X. Huang, A. Kteyan, S. X.-D. Tan, and V. Sukharev, "Physics-Based Electromigration Models and Full-Chip Assessment for Power Grid Networks," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 11, pp. 1848–1861, 2016.

[9] S. X.-D. Tan, H. Amrouch, T. Kim, Z. Sun, C. Cook, and J. Henkel, "Recent advances in EM and BTI induced reliability modeling, analysis and optimization," *Integration, the VLSI Journal*, vol. 60, pp. 132–152, Jan. 2018.

[10] C. Cook, Z. Sun, E. Demircan, M. D. Shroff, and S. X.-D. Tan, "Fast electromigration stress evolution analysis for interconnect trees using krylov subspace method," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 26, pp. 969–980, May 2018.

[11] S. Chatterjee, V. Sukharev, and F. N. Najm, "Power Grid Electromigration Checking using Physics-Based Models," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 7, pp. 1317–1330, 2018.

[12] H. Zhao and S. X.-D. Tan, "Postvoiding fem analysis for electromigration failure characterization," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 26, pp. 2483–2493, Nov. 2018.

[13] S. Feng, S. Gupta, A. Ansari, and S. Mahlke, "Maestro: Orchestrating lifetime reliability in chip multiprocessors," in *Proceedings of the 5th International Conference on High Performance Embedded Architectures and Compilers*, HiPEAC'10, (Berlin, Heidelberg), pp. 186–200, Springer-Verlag, 2010.

[14] A. Das, A. Kumar, and B. Veeravalli, "Reliability-driven task mapping for lifetime extension of networks-on-chip based multiprocessor systems," in *Proceedings of the Conference on Design, Automation and Test in Europe*, DATE '13, (San Jose, CA, USA), pp. 689–694, EDA Consortium, 2013.

[15] T. Kim, X. Huang, V. S. H.-B. CHen, and S. X.-D. Tan, "Learning-based dynamic reliability management for dark silicon processor considering EM effects," in *Proc. Design, Automation and Test In Europe Conf. (DATE)*, pp. 463–468, IEEE, March 2016.

[16] T. Kim, Z. Sun, H.-B. Chen, H. Wang, and S. X.-D. Tan, "Energy and lifetime optimizations for dark silicon manycore microprocessor considering both hard and soft errors," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 9, pp. 2561–2574, 2017.

[17] L. Huang, F. Yuan, and Q. Xu, "On Task Allocation and Scheduling for Lifetime Extension of Platform-Based MPSoC Designs," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, no. 12, pp. 2088–2099, 2011.

[18] T. Kim, B. Zheng, H.-B. Chen, Q. Zhu, V. Sukharev, and S. X.-D. Tan, "Lifetime optimization for real-time embedded systems considering electromigration effects," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, pp. 434–439, IEEE, Nov. 2014.

[19] Y. Ma, T. Chantem, R. P. Dick, and X. S. Hu, "Improving system-level lifetime reliability of multicore soft real-time systems," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, pp. 1895–1905, June 2017.

[20] Z. Sun, E. Demircan, M. D. Shroff, T. Kim, X. Huang, and S. X.-D. Tan, "Voltage-Based Electromigration Immortality Check for General Multi-Branch Interconnects," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, pp. 1–7, Nov. 2016.

[21] Z. Sun, E. Demircan, M. D. Shroff, C. Cook, and S. X.-D. Tan, "Fast Electromigration Immortality Analysis for Multisegment Copper Interconnect Wires," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 12, pp. 3137–3150, 2018.

[22] T. E. Carlson, W. Heirman, and L. Eeckhout, "Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulations," in *International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, pp. 52:1–52:12, Nov. 2011.

[23] V. Sukharev, A. Kteyan, and X. Huang, "Postvoiding stress evolution in confined metal lines," *IEEE Transactions on Device and Materials Reliability*, vol. 16, no. 1, pp. 50–60, 2016.

[24] V. Sukharev, A. Kteyan, and X. Huang, "Post-Voiding Stress Evolution in Confined Metal Lines," *IEEE Trans. on Device and Materials Reliability*, vol. 16, no. 1, pp. 50–60, 2016.

[25] M. A. Korhonen, P. Borgesen, D. D. Brown, and C.-Y. Li, "Microstructure based statistical model of electromigration damage in confined line metallizations in the presence of thermally induced stresses," *Journal of Applied Physics*, vol. 74, no. 8, pp. 4995–11, 1993.

[26] S. M. Alam, C. L. Gan, C. V. Thompson, and D. E. Troxel, "Reliability computer-aided design tool for full-chip electromigration analysis and comparison with different interconnect metallizations," *Microelectronics Journal*, vol. 38, no. 4, pp. 463–473, 2007.

[27] E. Demircan and M. D.Shroff, "Model based method for electro-migration stress determination in interconnects," in *2014 IEEE International Reliability Physics Symposium*, pp. IT.5.1–IT.5.6, June 2014.

[28] V. Sukharev, "Physically Based Simulation of Electromigration-Induced Degradation Mechanisms in Dual-Inlaid Copper Interconnects," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, pp. 1326–1335, Sept. 2005.

[29] A. Kteyan, V. Sukharev, M. A. Meyer, E. Zschech, and W. D. Nix, "Microstructure Effect on EM-Induced Degradations in Dual-Inlaid Copper Interconnects," *Proc. AIP Conference*, vol. 945, pp. 42–55, 2007.

[30] H. Chen, S. X.-D. Tan, V. Sukharev, X. Huang, and T. Kim, "Interconnect reliability modeling and analysis for multi-branch interconnect trees," in *Proc. Design Automation Conf. (DAC)*, pp. 1–6, IEEE, June 2015.

[31] C. Cook, Z. Sun, T. Kim, and S. X.-D. Tan, "Finite difference method for electromigration analysis of multi-branch interconnects," in *Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, pp. 1–4, IEEE, June 2016.

[32] M. Lin and A. Oates, "An electromigration failure distribution model for short-length conductors incorporating passive sinks/reservoirs," *IEEE Transactions on Device and Materials Reliability*, vol. 13, pp. 322–326, March 2013.

[33] R. Gleixner and W. Nix, "A Physically Based Model of Electromigration and Stress-Induced Void Formation in Microelectronic Interconnects," *Journal of Applied Physics*, vol. 86, no. 4, pp. 1932–1944, 1999.

[34] X. Wang, H. Wang, J. He, S. X.-D. Tan, Y. Cai, and S. Yang, "Physics-based Electromigration Modeling and Assessment for Multi-Segment Interconnects in Power Grid Networks," in *Proc. Design, Automation and Test In Europe Conf. (DATE)*, pp. 1727–1732, Mar. 2017.

[35] "Comsol multiphysics." https://www.comsol.com/ [Oct. 16, 2013].

[36] F. L. Wei, C. L. Gan, T. L. Tan, C. S. Hau-Riege, A. P. Marathe, J. J. Vlassak, and C. V. Thompson, "Electromigration-induced extrusion failures in cu/low-k interconnects," *Journal of Applied Physics*, vol. 104, no. 023529, pp. 1–10, 2008.

[37] C. S. Hau-Riege, A. P. Marathe, and Z. S. Choi, "The effect of current direction on the electromigration in short-lines with reservoirs," in *IEEE International Reliability Physics Symposium (IRPS)*, pp. 381–384, 2008.

[38] R. G. Filippi, R. A. Wachnik, H. Aochi, J. R. Lloyd, and M. A. Korhonen, "The effect of current density and stripe length on resistance saturation during electromigration testing," *Applied Physics Letters*, vol. 69, pp. 2350–2352, Oct. 1996.

[39] C. M. Tan, *Electomigration in ULSI Interconnects*. International Series on Advances in Solid State Electronics and Technology, Word Scientific, 2010.

[40] V. Mishra and S. S. Sapatnekar, "Probabilistic wire resistance degradation due to electromigration in power grids," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, pp. 628–640, April 2017.

[41] C.-K. Hu, D. Canaperi, S. T. Chen, L. M. Gignac, B. Herbst, S. Kaldor, M. Krishnan, E. Liniger, D. L. Rath, D. Restaino, R. Rosenberg, J. Rubino, S.-C. Seo, A. Simon, S. Smith, and W.-T. Tseng, "Effects of overlayers on electromigration reliability improvement for cu/low k interconnects," in *Reliability Physics Symposium Proceedings, 2004. 42nd Annual. 2004 IEEE International*, pp. 222–228, IEEE, 2004.

[42] L. Zhang, *Effects of Scaling and Grain Structure on Electromigration Reliability of Cu Interconnects*. PhD thesis, University of Texas at Austin, 2010.

[43] T. N. Marieb, E. Abratowski, J. C. Bravman, M. Madden, and P. Flinn, "Direct observation of the growth and movement of electromigration voids under passivation," *AIP Conference Proceedings*, vol. 305, no. 1, pp. 1–14, 1994.

[44] H. Zhao and S. X.-D. Tan, "Multi-physics-based FEM analysis for post-voiding analysis of electromigration failure effects," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, pp. 1–8, IEEE, Nov. 2018.

[45] C. W. Chang, Z.-S. Choi, C. V. Thompson, C. L. Gan, K. L. Pey, W. K. Choi, and N. Hwang, "Electromigration resistance in a short three-contact interconnect tree," *Journal of Applied Physics*, vol. 99, no. 9, p. 094505, 2006.

[46] X. Huang, A. Kteyan, S. X.-D. Tan, and V. Sukharev, "Physics-based electromigration models and full-chip assessment for power grid networks," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, pp. 1848–1861, Nov. 2016.

[47] V. Sukharev and F. N. Najm, "Electromigration check: Where the design and reliability methodologies meet," *IEEE Transactions on Device and Materials Reliability*, vol. 18, pp. 498–507, Dec 2018.

[48] S. P. Hau-Riege and C. V. Thompson, "Experimental characterization and modeling of the reliability of interconnect trees," *Journal of Applied Physics*, vol. 89, no. 1, pp. 601–609, 2001.

[49] Z. Sun, S. Sadiqbatcha, H. Zhao, and S. X.-D. Tan, "Accelerating Electromigration Aging for Fast Failure Detection for Nanometer ICs," in *Proc. Asia South Pacific Design Automation Conf. (ASPDAC)*, pp. 623–630, Jan. 2018.

[50] A. Limited, "Arm cortex-m0 designstart," 2010.

[51] "Synopsys 32/28nm generic library for teaching ic design." http://www.synopsys.com.

[52] R. Goldman, K. Bartleson, T. Wood, K. Kranen, C. Cao, V. Melikyan, and G. Markosyan, "Synopsys' open educational design kit: Capabilities, deployment and future," in *2009 IEEE International Conference on Microelectronic Systems Education*, pp. 20–24, July 2009.

[53] R. L. de Orio, *Electromigration modeling and simulation*. na, 2010.

[54] "Engineering toobox." http://www.engineeringtoolbox.com.

[55] S. P. Hau-Riege and C. V. Thompson, "The Effects of The Mechanical Properties of The Confinement Material on Electromigration in Metallic Interconnects," *Journal of Materials Research*, vol. 15, no. 8, pp. 1797–1802, 2000.

[56] K. Tu, " Recent Advances on Electromigration in Very-Large-Scale-Integration of Interconnects," *Journal of Applied Physics*, vol. 94, no. 9, pp. 5451–5473, 2003.

[57] V. Sukharev, A. Kteyan, and E. Zschech, "Physics-Based Models for EM and SM Simulation in Three-Dimensional IC Structures," *IEEE Trans. on Device and Materials Reliability*, vol. 12, no. 2, pp. 272–284, 2012.

[58] V. Sukharev, A. Kteyan, J.-H. Choy, H. Hovsepyan, A. Markosian, E. Zschech, and R. Huebner, "Multi-scale Simulation Methodology for Stress Assessment in 3D IC: Effect of Die Stacking on Device Performance," *Journal of Electronic Testing*, vol. 28, pp. 63–72, Nov. 2011.

[59] S. X.-D. Tan, M. Tahoori, T. Kim, S. Wang, Z. Sun, and S. Kiamehr, *VLSI Systems Long-Term Reliability – Modeling, Simulation and Optimization*. Springer Publishing, 2019.

[60] R. G. Filippi, P.-C. Wang, A. Brendler, K. Chanda, and J. R. Lloyd, "Implications of a threshold failure time and void nucleation on electromigration of copper interconnects," *Journal of Applied Physics*, vol. 107, no. 10, 2010.

[61] V. Mishra and S. S. Sapatnekar, "Predicting Electromigration Mortality Under Temperature and Product Lifetime Specifications," in *Proc. Design Automation Conf. (DAC)*, pp. 1–6, Jun. 2016.

[62] A. Abbasinasab and M. Marek-Sadowska, "RAIN: A tool for reliability assessment of interconnect networks—physics to software," in *Proc. Design Automation Conf. (DAC)*, (New York, NY, USA), pp. 133:1–133:6, ACM, 2018.

[63] Z. Lu, W. Huang, J. Lach, M. Stan, and K. Skadron, "Interconnect lifetime prediction under dynamic stress for reliability-aware design," in *Proc. IEEE/ACM Int. Conf. on Computer Aided Design (ICCAD)*, pp. 327–334, IEEE, November 2004.

[64] D. Zhu, R. Melhem, and D. Mosse, "The effects of energy management on reliability in real-time embedded systems," in *Proceedings of the 2004 IEEE/ACM International Conference on Computer-aided Design*, ICCAD '04, (Washington, DC, USA), pp. 35–40, IEEE Computer Society, 2004.

[65] L. Tan, S. Song, P. Wu, Z. Chen, R. Ge, and D. Kerbyson, "Investigating the interplay between energy efficiency and resilience in high performance computing," in *Parallel and Distributed Processing Symposium (IPDPS), 2015 IEEE International*, pp. 786–796, May 2015.

[66] Y. Zhang, K. Chakrabarty, and V. Swaminathan, "Energy-aware fault tolerance in fixed-priority real-time embedded systems," in *Computer Aided Design, 2003. ICCAD-2003. International Conference on*, pp. 209–213, Nov 2003.

[67] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*. Cambridge, MA, USA: MIT Press, 1st ed., 1998.

[68] M. L. Littman, T. L. Dean, and L. P. Kaelbling, "On the complexity of solving markov decision problems," in *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, (San Francisco, CA, USA), pp. 394–402, Morgan Kaufmann Publishers Inc., 1995.

[69] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller, "Playing atari with deep reinforcement learning," *CoRR*, vol. abs/1312.5602, 2013.

[70] C. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[71] S. Woo, M. Ohara, E. Torrie, J. Singh, and A. Gupta, "The splash-2 programs: characterization and methodological considerations," in *Computer Architecture, 1995. Proceedings., 22nd Annual International Symposium on*, pp. 24–36, June 1995.

[72] J. H. Ahn, S. Li, O. Seongil, and N. Jouppi, "Mcsima+: A manycore simulator with application-level+ simulation and detailed microarchitecture modeling," in *Performance Analysis of Systems and Software (ISPASS), 2013 IEEE International Symposium on*, pp. 74–85, April 2013.

[73] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi, "Mcpat: An integrated power, area, and timing modeling framework for multicore and manycore architectures," in *2009 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 469–480, Dec 2009.

[74] K.Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature-aware microarchitecture," in *International Symposium on Computer Architecture*, pp. 2–13, 2003.

[75] Abad *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, OSDI'16, (Berkeley, CA, USA), pp. 265–283, USENIX Association, 2016.

[76] T. Kolpe, A. Zhai, and S. Sapatnekar, "Enabling improved power management in multicore processors through clustered dvfs," in *Proc. Design, Automation and Test In Europe. (DATE)*, pp. 1–6, March 2011.

[77] K. Bernstein, D. J. Frank, A. E. Gattiker, W. Haensch, B. L. Ji, S. R. Nassif, E. J. Nowak, D. J. Pearson, and N. J. Rohrer, "High-performance cmos variability in the 65-nm regime and beyond," *IBM Journal of Research and Development*, vol. 50, pp. 433–449, July 2006.

[78] S. Chatterjee, M. B. Fawaz, and F. N. Najm, "Redundancy-Aware Electromigration Checking for Mesh Power Grids," in *Proc. IEEE/ACM Int. Conf. on Computer Aided Design (ICCAD)*, pp. 540–547, 2013.

[79] W. Wang, S. Yang, S. Bhardwaj, S. Vrudhula, F. Liu, and Y. Cao, "The impact of nbti effect on combinational circuit: Modeling, simulation, and analysis," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, pp. 173–183, Feb 2010.

[80] F. Oboril and M. B. Tahoori, "Extratime: Modeling and analysis of wearout due to transistor aging at microarchitecture-level," in *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2012)*, pp. 1–12, June 2012.

[81] K.-L. Chen, S. Saller, and R. Shah, "The case of ac stress in the hot-carrier effect," *IEEE Transactions on Electron Devices*, vol. 33, pp. 424–426, Mar 1986.

[82] E. Takeda, Y. Nakagome, H. Kume, and S. Asai, "New hot-carrier injection and device degradation in submicron mosfets," *IEE Proceedings I - Solid-State and Electron Devices*, vol. 130, pp. 144–150, June 1983.

[83] K. A. Bowman, B. L. Austin, J. C. Eble, X. Tang, and J. D. Meindl, "A physical alpha-power law mosfet model," in *Proceedings of the 1999 International Symposium on Low Power Electronics and Design*, ISLPED '99, pp. 218–222, 1999.

[84] J. Srinivasan, S. V. Adve, P. Bose, and J. A. Rivers, "The case for lifetime reliability-aware microprocessors," in *Proceedings. 31st Annual International Symposium on Computer Architecture, 2004.*, pp. 276–287, June 2004.

[85] C. Bolchini, M. Carminati, M. Gribaudo, and A. Miele, "A lightweight and open-source framework for the lifetime estimation of multicore systems," in *2014 IEEE 32nd International Conference on Computer Design (ICCD)*, pp. 166–172, Oct 2014.

[86] C. Lin, C. Chang, Y. Syu, J. Wu, P. Liu, P. Cheng, and W. Hsu, "An energy-efficient task scheduler for multi-core platforms with per-core dvfs based on task characteristics," in *2014 43rd International Conference on Parallel Processing*, pp. 381–390, Sep. 2014.

[87] A. Limited, "Arm cortex-a75 technical reference manual," 2016.

[88] C. Liu and J. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *Journal of the ACM*, vol. 20, no. 1, pp. 46–61, 1973.

[89] N. Suzuki, H. Kim, D. de Niz, B. Andersson, L. Wrage, M. Klein, and R. R. Rajkumar, "Coordinated bank and cache coloring for temporal protection of memory accesses," in *IEEE International Conference on Embedded Software and Systems (ICESS)*, 2013.

[90] H. Kim, D. de Niz, B. Andersson, M. Klein, O. Mutlu, and R. Rajkumar, "Bounding and reducing memory interference in COTS-based multi-core systems," *Real-Time Systems*, vol. 52, no. 3, pp. 356–395, 2016.

[91] H. Kim, A. Kandhalu, and R. R. Rajkumar, "A coordinated approach for practical OS-level cache management in multi-core real-time systems," in *Euromicro Conference on Real-Time Systems (ECRTS)*, 2013.

[92] H. Kim and R. Rajkumar, "Real-time cache management for multi-core virtualization," in *ACM International Conference on Embedded Software (EMSOFT)*, 2016.

[93] H. Yun, G. Yao, R. Pellizzoni, M. Caccamo, and L. Sha, "Memory access control in multiprocessor for real-time systems with mixed criticality," in *Euromicro Conference on Real-Time Systems (ECRTS)*, 2012.

[94] H. Yun, R. Mancuso, Z.-P. Wu, and R. Pellizzoni, "PALLOC: DRAM bank-aware memory allocator for performance isolation on multicore platforms," in *IEEE Real-Time Technology and Applications Symposium (RTAS)*, 2014.

[95] R. Mancuso, R. Dudko, E. Betti, M. Cesati, M. Caccamo, and R. Pellizzoni, "Real-time cache management framework for multi-core architectures," in *IEEE Real-Time Technology and Applications Symposium (RTAS)*, 2013.

[96] D. S. Johnson, A. Demers, J. D. Ullman, M. R. Garey, and R. L. Graham, "Worst-case performance bounds for simple one-dimensional packing algorithms," *SIAM Journal on Computing*, vol. 3, no. 4, pp. 299–325, 1974.

[97] Y. Ma, T. Chantem, X. S. Hu, and R. P. Dick, "Improving lifetime of multicore soft real-time systems through global utilization control," in *Proceedings of the 25th Edition on Great Lakes Symposium on VLSI*, GLSVLSI '15, (New York, NY, USA), pp. 79–82, ACM, 2015.

[98] A. K. Mok, "Fundamental design problems of distributed systems for the hard real-time environment," *PhD Thesis, Massachusetts Institute of Technology*, 1983.

[99] M. Joseph and P. K. Pandya, "Finding response times in a real-time system," *The Computer Journal*, vol. 29, no. 5, pp. 390–395, 1986.

[100] D. Brodowski and N. Golde, "Linux cpufreq governors," *Linux Kernel. https://www. kernel. org/doc/Documentation/cpu-freq/governors. txt*, 2013.

[101] V. Pallipadi, "Enhanced intel speedstep technology and demand-based switching on linux," *Intel Developer Service*, 2009.

[102] M. Mandelli, G. Castilhos, G. Sassatelli, L. Ost, and F. G. Moraes, "A distributed energy-aware task mapping to achieve thermal balancing and improve reliability of many-core systems," in *Proceedings of the 28th Symposium on Integrated Circuits and Systems Design*, SBCCI '15, (New York, NY, USA), pp. 13:1–13:7, ACM, 2015.

[103] A. Naithani, S. Eyerman, and L. Eeckhout, "Reliability-aware scheduling on heterogeneous multicore processors," in *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pp. 397–408, Feb 2017.

[104] J. Sun, R. Lysecky, K. Shankar, A. Kodi, A. Louri, and J. Roveda, "Workload assignment considering NBTI degradation in multicore systems," *J. Emerg. Technol. Comput. Syst.*, vol. 10, pp. 4:1–4:22, Jan. 2014.